An ℓ_1 Penalty Function Approach to the Nonlinear Bilevel Programming Problem

by

Lori Michelle Case

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 1997

0-612-21334-X

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The nonlinear bilevel programming problem is a constrained optimization problem defined over two vectors of unknowns, x and y. Feasibility constraints on (x, y) include the requirement that y is a solution of another optimization problem, called the inner problem, which is parameterized by x. The bilevel problem is very difficult to solve, and few algorithms have been published for the nonlinear problem. Therefore, instead of solving the bilevel problem directly, a "simpler", related problem is solved. This problem is defined by replacing the solution constraint in the bilevel problem with a set of conditions which must be satisfied at a minimum point of the inner problem. The resulting one level mathematical program is solved using an exact penalty function technique, which involves finding solutions to a series of unconstrained problems. These problems are usually nonconvex and nondifferentiable. Each problem is solved within a trust region framework, and specialized techniques are developed to overcome difficulties due to the nondifferentiabilities. A unique approach is developed to resolve degeneracy in the penalty function problems. The algorithm is proven to converge to a minimum point of the penalty function. Testing results are presented and analyzed.

Acknowledgements

I am grateful for the financial support provided by NSERC, the Government of Ontario, the University of Waterloo, and ITRC/ICR.

I am very grateful to my supervisors, Dr. Paul Calamai of the Department of Systems Design Engineering, and Dr. Andrew Conn of T.J. Watson Research Center, New York, for their help over the course of my research. I thank them especially for their faith in me, and for their ability to help me believe in myself.

Special thanks go to my family and friends for their love over these many years. I would most like to thank my husband, Greg, whose friendship, support and strength has been unwavering and much needed, and my daughter, Ellen, for motivating me to complete this work.

This work is dedicated to the memory of my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The bilevel programming problem is a constrained optimization problem defined over two vectors of unknowns, x and y. Among the constraints is a requirement that y is a solution of another optimization problem, called the inner or lower level problem. The inner problem, which is defined over both sets of unknowns, is parameterized by x and is optimized only with respect to y. Thus, the bilevel programming problem involves two connected hierarchical levels of optimization.

Mathematically, the bilevel problem can be described as follows for $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, where n and m are positive integers. Define $P = \{1, \cdots, p\}$ and $T = \{1, \cdots, t\}$ for nonnegative integers p and t.

$$\text{BP}: \quad \min_{x,y} \quad F(x, y) \quad \text{subject to} \quad G_i(x, y) \geq 0,\ i \in T, \text{ and } y \in R(x),$$

where

$$R(x) \quad = \quad \{y \in \mathbb{R}^m : y \text{ solves LLP}(x)\},$$

and

$$\text{LLP}(x): \quad \min_{y} \quad f(x, y) \quad \text{subject to} \quad g_i(x, y) \geq 0,\ i \in P.$$

Assume that all functions in x and y are continuous and twice differentiable over $\mathbb{R}^{n+m}$.

A point (x, y) is feasible for BP if it satisfies $G_i(x, y) \geq 0$ for all $i \in T$, as well as the solution constraint $y \in R(x)$. Note that feasibility of (x, y) for BP implies the feasibility of y for LLP(x). No assumption is made regarding the existence or uniqueness of a solution of LLP(x). If LLP(x) has no solution for some x, then there exists no y such that (x, y) is feasible for BP. For a given x, $R(x)$ is a multiple-valued function of x. The set may be empty, contain a single element, or contain more than one element. Thus, at a feasible point (x, y), y is an implicit function of x.

A global solution of BP is a feasible point which minimizes the value of the objective function $F(x, y)$ over the set of all feasible points of BP. The bilevel problem is usually nonconvex as a result of the solution constraint. Consequently, bilevel problems often have local solutions in addition to any global solutions.

Most research to date has concentrated on the linear bilevel problem, in which all the functions are linear over $\mathbb{R}^{n+m}$. Even in this restricted form, the problem is often very difficult to solve due to the solution constraint and the resulting nonconvex feasible region. Ben-Ayed and Blair [15] proved that the linear bilevel problem is NP-hard. Nonlinearity in the objective and constraint functions complicates matters significantly. While there exist algorithms for the nonlinear case of bilevel programming, to date all techniques for which extensive numerical results have been presented have assumed special forms for the bilevel problems being solved (including, for example, linear constraints, convex problems, or separable quadratic objective functions). This thesis describes a theoretical and practical algorithm for solving nonlinear bilevel problems, and presents extensive numerical results along with convergence results.

Chapter 2 presents a more detailed description of the bilevel problem, including some applications and properties. It also includes a description of several approaches which have been used to solve bilevel problems.

Due to the difficulty in directly solving the nonlinear bilevel problem, the proposed algorithm instead solves a related problem. By replacing the solution constraint with a set of conditions that must be satisfied by a solution of LLP(x), a one level problem is defined. This related problem is actually equivalent to the bilevel problem under a stated set of conditions. However, these conditions are not usually satisfied. The related problem, like the bilevel problem, is generally nonconvex, and it may be difficult to find an initial feasible point. Therefore, an ℓ_1 exact penalty function is used to solve the related problem. This technique involves solving a series of unconstrained optimization problems. Chapter 3 defines the related problem and describes the penalty function algorithm. It also includes a description of the trust region technique used to solve the unconstrained penalty function problems. Chapter 4 focuses on the resolution of several problems which arise when the theoretical algorithm of Chapter 3 is actually implemented. In Chapter 5, the algorithm is proven to converge to a minimum point of the penalty function.

As a result of the structure of the related problem, standard degeneracy resolving techniques proved inappropriate. Chapter 6 describes the nature of degeneracy in the penalty function and details the method developed to recover from the difficulties it causes. The technique, which is one of the major contributions of this thesis, is also proven to work.

The algorithm implementation was tested on a variety of bilevel problems found in the literature. In addition, it was tested on some larger bilevel problems which were generated by the technique of Calamai and Vicente [28], and on some original nonlinear problems. The algorithm was quite successful in identifying local solutions of both linear and nonlinear bilevel problems. Detailed results and analysis are presented in Chapter 7. A listing of the test problems is included in Appendix A.

This dissertation concludes with Chapter 8, which summarizes the major contributions of this research and discusses possible future work. Appendix B includes a listing of some of the notation used throughout this work.

Chapter 2

Bilevel Programming

This chapter provides an overview of bilevel programming. Needed terminology is introduced in Section 2.1, followed in the next two sections by descriptions of applications and of problems related to bilevel programming. Two example problems are discussed in Section 2.4, and some general characteristics of bilevel programming are presented in Section 2.5. The chapter concludes with a summary of existing methods for solving bilevel problems and a brief overview of the proposed technique.

2.1 Introduction

The following definitions are required. Note that the notation of the introduction is continued.

Definition 2.1

1. *The outer or upper level problem is*

$$ULP: \min_{x,y} \; F(x,y) \quad \textit{subject to} \quad G_i(x,y) \geq 0, \; i \in T.$$

2. *The inner or lower level problem, parameterized by x, is*

$$LLP(x): \min_{y} \; f(x,y) \quad \text{subject to} \quad g_i(x,y) \geq 0, \; i \in P.$$

3. *$R(x)$, the set of solutions of $LLP(x)$, is called the rational reaction set of x. It contains both local and global solutions of the lower level problem*[1].

4. *The solution constraint on y is the constraint $y \in R(x)$.*

5. *The induced region is the set of feasible points of the bilevel problem.*

$$\mathcal{I} = \{(x,y) \in \mathbb{R}^{n+m} : G_i(x,y) \geq 0 \text{ for } i \in T, \text{ and } y \in R(x)\}.$$

6. *The reduced or relaxed problem is*

$$\begin{aligned} RBP: \min_{x,y} \; F(x,y) \quad \text{subject to} \quad & G_i(x,y) \geq 0, \quad i \in T \\ & g_i(x,y) \geq 0, \quad i \in P. \end{aligned}$$

7. *A solution of the bilevel problem is a solution of*

$$\min_{x,y} F(x,y) \text{ subject to } (x,y) \in \mathcal{I}.$$

This definition includes both local and global solutions of the problem.

2.2 Applications

The earliest explicit use of the bilevel problem as denoted by BP in Chapter 1 dates from 1977, in the work of Candler and Norton [30]. The authors use a linear bilevel problem to simulate the relationship between the actions of the government and farmers

[1] Some definitions of bilevel problems (for example, [7], [13], and [67]) include an assumption that the inner problem LLP(x) is convex and has a unique global solution for each feasible value of x. We feel that the nonlinearity of LLP(x) warrants the inclusion of its local solutions in $R(x)$.

in a Mexican agricultural system. The upper level variables model various decisions made by the government: for example, levels of subsidies on fertilizer, prices for different crops, and water taxes. The lower level variables correspond to the actions of the farmers, and include the amount of fertilizer and water used, and the amounts and types of crops planted. The government attempts to achieve some goal (for example, minimizing their expenditures or maximizing production of various crops). At the same time, the farmers attempt to maximize their own profits. Candler and Norton note that problems of this type were traditionally solved using one level linear programs. However, such models do not truly reflect the dependencies within the relationship between the actions of the two parties, particularly when they have conflicting goals.

Bilevel programming has since been used to solve other problems from the field of economics (see [29] and [43]). Many other sources have yielded applications as well. Two examples are noted below.

- Network Design Problems. In [54], LeBlanc and Boyce use a bilevel problem to model a network of roads, as an aid in deciding which roads should be upgraded in order to improve network performance. Some roads can be added or improved in an attempt to reduce network congestion, but the drivers who use the network will always attempt to achieve their goal (for example, to take the shortest route between two points). Proposed improvements of various routes are reflected in the upper level variables, and the lower level variables correspond to the actual routes taken by the network users.

  Other authors (see [16] and [56]) have also considered this type of application.

- Environmental Policy. Kolstad [52] describes how bilevel programming can be used to model environmental regulation. In order to achieve pollution concentration standards while minimizing social costs, the government sets emissions taxes on polluters from industry. The tax levels are the upper level variables. Industry may react in many ways in order to maximize profits, and these reactions correspond to the lower level variables. Industry's level of emissions may increase or remain unchanged if the tax level is too low, or it may decrease if the tax is restrictive. However, if the tax is too high, industry cutbacks and a subsequent increase in unemployment may result.

In [9], Bard describes another application of bilevel programming in evaluating environmental policy.

The reader is referred to Anandalingham and Friesz [7], Kolstad [52], Nicholls [58], and Vicente and Calamai [66] for more examples of applications of bilevel programming.

2.3 Related Problems

While research into the bilevel problem BP is quite recent, there has been active research for many years into equivalent and related problems.

The continuous max-min problem (for example, see [42]) is a special case of BP, in which the upper and lower level objective functions satisfy $f(x, y) = -F(x, y)$. This problem has been studied in many forms (linear and nonlinear, unconstrained and constrained). Techniques used to solve max-min problems will not generally be applicable to bilevel problems. However, an algorithm which solves BP can also solve max-min problems.

Static constrained Stackelberg problems (defined as duopoly problems in [68]), which model two person games as optimization problems, are actually bilevel problems. Much of the literature (for example, [2], [6], and [59]) on bilevel problems refers to the problems being studied as Stackelberg problems.

Large scale linear problems are sometimes solved using decomposition techniques (see [27]). These techniques involve identifying sets of variables of the linear problem which do not interact with each other. Subproblems are defined for each of these sets of variables. The objective functions of the subproblems are derived from the objective function of the original problem. These problems look very similar to bilevel problems: the reduced (lower level) problems are solved, and are then combined into a single solution by the original (upper level) problem. However, as noted in [29] and [52], there are several important distinctions between decomposition problems and true bilevel problems. All the variables are actually controlled by one entity in the decomposition problems, unlike the bilevel problems. Also, while the objective functions in BP may be conflicting or noncooperative, the objective functions in the decomposition problems are always cooperative since they are derived from the single objective function of the original problem. However, the relationship between the decomposition techniques and bilevel problems indicate that a general technique for bilevel programming might be useful in solving some large scale nonlinear optimization problems. Conversely, a decomposition technique could serve as motivation for a new algorithm for bilevel programming. In fact, in [5], Alexandrov and Dennis describe a decomposition technique for solving nonlinear programs, and in [4], they present a theoretical framework for solving unconstrained bilevel problems based on their earlier work.

Another problem which is closely related to BP was posed in the early 1970's, by Bracken and McGill ([23], [24], [25], and [26]) and Geoffrion and Hogan [45]. These problems are called optimal value bilevel problems, and can be written as follows.

$$\text{OVBP}: \min_x \quad F(x, v(x)) \quad \text{subject to} \quad G_i(x, v(x)) \geq 0, \quad i \in T,$$

where

$$v(x) \;=\; \min_y \quad f(x,y) \quad \text{subject to} \quad g_i(x,y) \geq 0, \quad i \in P.$$

Note that the optimal value of the lower level problem is referenced in the upper level functions, not the solution of the lower problem. As a result of this difference, OVBP has one significant property that BP, in general, does not. Bracken and McGill [24] verified that if the upper and lower level problems are both convex, then OVBP is also convex. Consequently, optimal value bilevel problems are usually easier to solve than bilevel problems like BP.

The generalized bilevel problem (see [7]) is an extension of the bilevel problem BP, in which LLP(x) is replaced by a variational inequality (that is, an infinite set of constraints), as stated below.

$$\begin{aligned} \text{GBP}: \quad & \min_{x,y} \quad F(x,y) \\ & \text{s.t.} \quad x \in X,\ y \in Y \\ & \text{and} \quad f(x,y)(y'-y) \geq 0,\ \forall y' \in Y. \end{aligned}$$

Under appropriate assumptions, the variational inequality is equivalent to a mathematical program. Therefore, any technique to solve GBP can also be used to solve BP. However, because not every variational inequality corresponds to a mathematical program, techniques designed for solving BP may not be appropriate for solving GBP.

2.4 Example Problems

Two small bilevel problems are presented to illustrate some properties of bilevel programs.

Consider the following bilevel problem.

$$\text{BP}_1: \quad \min_{x,y} \quad (x+1)y \quad \text{subject to} \quad 0 \leq x \leq 2 \text{ and } y \in R_1(x),$$

where

$$R_1(x) = \{y \in \mathbb{R} : y \text{ solves } \text{LLP}_1(x)\}$$

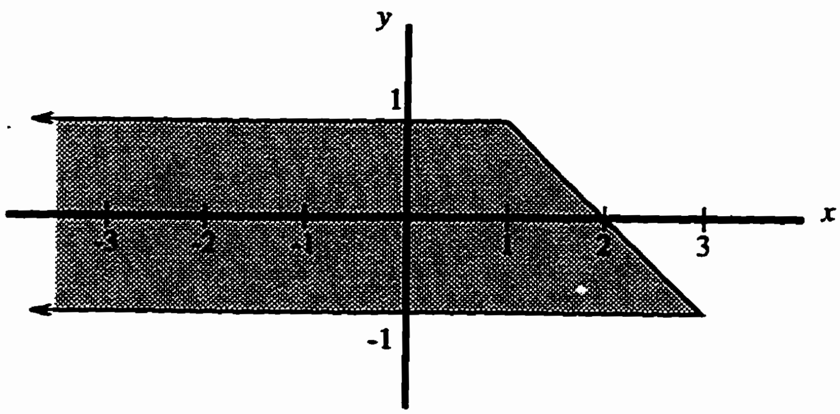

Figure 2.1: $\rho_1(x)$ (shaded region), for $x \in \mathbb{R}$

and

$$\text{LLP}_1(x): \min_y \; x - y \quad \text{subject to} \quad x + y \leq 2 \text{ and } -1 \leq y \leq 1.$$

To determine the induced region of BP_1, we must examine the lower level problem carefully. Let $\rho_1(x)$ be the feasible region of $\text{LLP}_1(x)$, for any value of x. The set is illustrated in Figure 2.1 and is described mathematically as,

$$\rho_1(x) = \begin{cases} [-1, 1] & \text{if } x \leq 1 \\ [-1, 2 - x] & \text{if } 1 < x \leq 3 \\ \emptyset & \text{if } 3 < x. \end{cases}$$

Now, $\text{LLP}_1(x)$ is simply

$$\min_y \; -y \quad \text{subject to} \quad y \in \rho_1(x).$$

The argmin solution of this problem,

$$R_1(x) = \begin{cases} \{1\} & \text{if } x \leq 1 \\ \{2 - x\} & \text{if } 1 < x \leq 3 \\ \emptyset & \text{otherwise,} \end{cases}$$

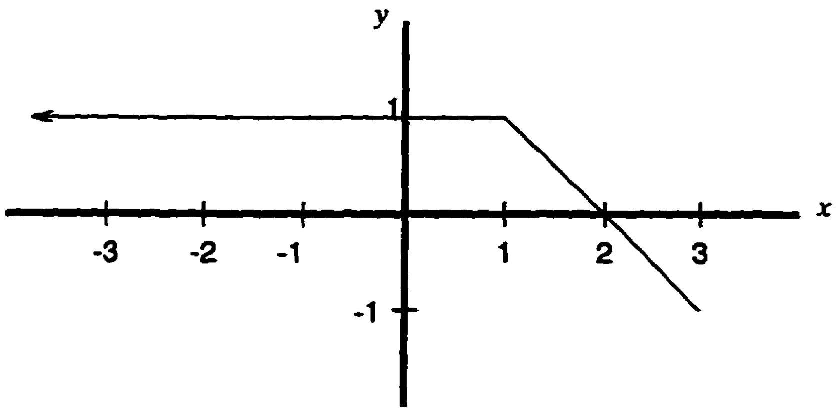

Figure 2.2: $R_1(x)$ (connected line segments), for $x \in \mathbb{R}$

is the rational reaction set of the bilevel problem BP_1, for $x \in \mathbb{R}$, and is shown in Figure 2.2. With this simplification, the feasible or induced region of the bilevel problem is

$$\mathcal{I}_1 = \{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq 2 \text{ and } y \in R_1(x)\},$$

as illustrated by the two connected line segments in Figure 2.3. Note that even with a univariate, convex linear problem as the lower level problem, the induced region of BP_1 is not convex. Therefore, BP_1, now stated,

$$\min_{x,y} \quad (x+1)y \quad \text{subject to } (x, y) \in \mathcal{I}_1$$

is nonconvex and may have multiple local minima. Figure 2.4 displays several contour lines of the objective function along with $\mathcal{I}$. From the diagram, it is clear that feasible descent is possible from any point along the line segments, except $A = (0, 1)$ and $B = (2, 0)$. Therefore, both A and B are local solutions of BP_1. Because the objective function value is lower at B than A, the point B is the global solution of BP_1.

Note that rewriting BP_1 by moving a lower level constraint to the upper level of the bilevel problem significantly changes the structure of the bilevel problem. Consider

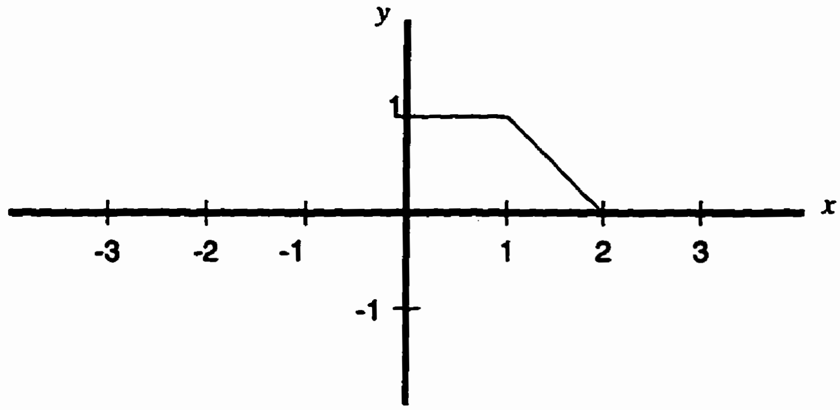

Figure 2.3: $\mathcal{I}_1$ (connected line segments), Feasible region of BP_1

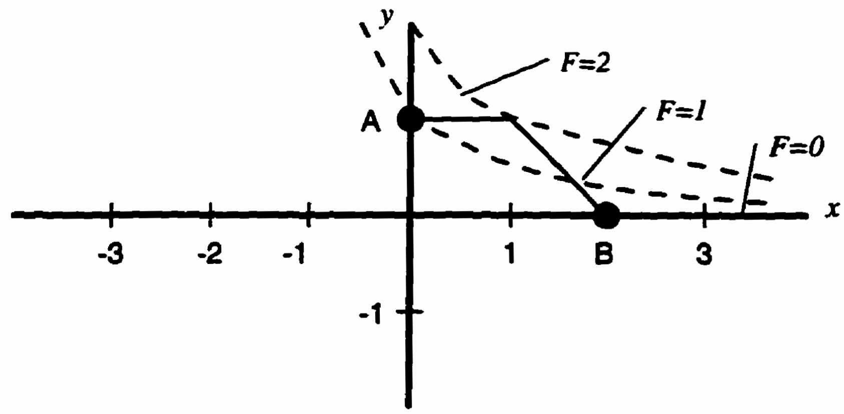

Figure 2.4: Contours (dashed lines) of the Objective Function

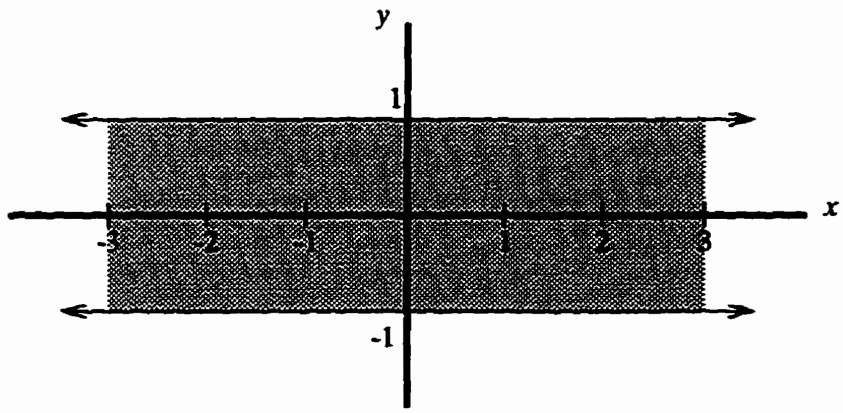

Figure 2.5: $\rho_2(x)$ (shaded region) and $R_2(x)$ $(y = 1)$, for $x \in I\!R$

moving the lower level constraint $x + y \leq 2$.

$$\text{BP}_2: \min_{x,y} \quad (x+1)y \quad \text{subject to} \quad x+y \leq 2,\ 0 \leq x \leq 2, \ \text{and}\ y \in R_2(x),$$

where

$$R_2(x) = \{y \in I\!R : y \text{ solves } \text{LLP}_2(x)\}$$

and

$$\text{LLP}_2(x): \min_{y} \quad x - y \quad \text{subject to} \quad -1 \leq y \leq 1.$$

Using similar analysis for BP_2 as used above for BP_1, $\rho_2(x)$, the feasible region of $\text{LLP}_2(x)$, and $R_2(x)$, the rational reaction set of $\text{LLP}_2(x)$ satisfy

$$\begin{aligned} \rho_2(x) &= \{y \in I\!R : -1 \leq y \leq 1\} \\ R_2(x) &= \{1\}, \end{aligned}$$

for all $x \in I\!R$. These sets are illustrated in Figure 2.5. The feasible region, $\mathcal{I}_2$, for BP_2 (as shown by the solid line segment in Figure 2.6) is significantly different from $\mathcal{I}_1$.

$$\mathcal{I}_2 = \{(x,y) : 0 \leq x \leq 2,\ x+y \leq 2,\ y \in R_2(x)\}$$

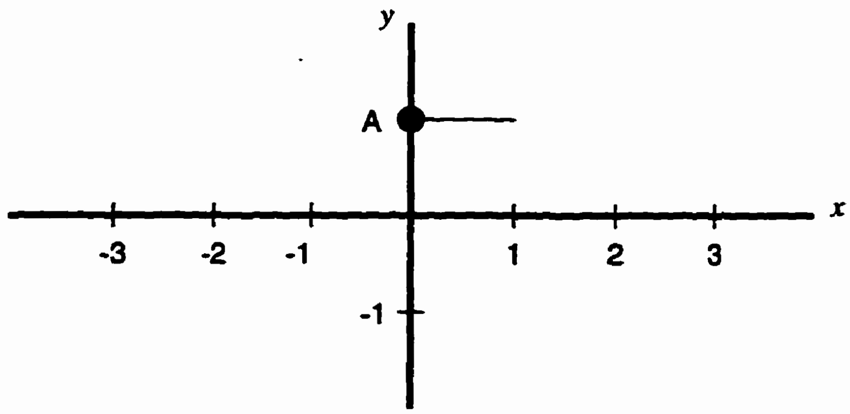

Figure 2.6: $\mathcal{I}_2$ (solid line segment), Feasible Region of BP_2

$$\begin{aligned}\mathcal{I}_2 &= \{(x,y) : 0 \leq x \leq 2,\ x+y \leq 2,\ y = 1\} \\ &= \{(x,y) : 0 \leq x \leq 1,\ y = 1\}.\end{aligned}$$

The set $\mathcal{I}_2$ is convex and BP_2 is a convex bilevel problem. Note that the global solution of BP_1, is not in the induced region of BP_2, and therefore is not even feasible for BP_2.

Recalling the contours of the objective function shown in Figure 2.4, the point A, a local solution of BP_1, is the global solution of BP_2.

Even though the same functions define both BP_1 and BP_2, the problems are significantly different and have different solution sets.

2.5 Characteristics of the Bilevel Problem

Some of the properties exhibited by the example bilevel problems BP_1 and BP_2 are characteristic of bilevel problems in general, while others are not.

- BP_1 is not convex, despite the convexity of $LLP_1(x)$ for all feasible x. While BP_2 is a convex problem, most bilevel problems are not convex due to the solution constraint.

- The solutions of BP_1 and BP_2 occur at vertices of the feasible region of the reduced problem. All linear bilevel problems have this property, and, as will be seen in Section 2.6.1, it is the basis for many algorithms designed to solve this special case of bilevel programming. However, this property is not usually present in nonlinear problems, for which there may be no vertices in the feasible region.

- The rational reaction sets of LLP_1 and LLP_2, for feasible values of x, are singletons corresponding to unique global solutions of the lower level problems. This is not generally the case for a nonlinear lower level problem, which may have multiple global and local minima. Combining this fact with the first property above, local solutions are possible at both levels of the problem.

  Some definitions of the bilevel problem (for example, [62]) assume that the solution set of the lower level problem is a singleton for all feasible x. We consider that to be a very strong assumption, and it is not made here. Consequently, we must acknowledge the possibility of local solutions.

- Although not evidenced by the problems BP_1 and BP_2, another concern in solving any optimization problem is the possibility of degeneracy. As with the issue of local solutions, degeneracy may be present at both levels of the bilevel problem. Degeneracy is the focus of Chapter 6.

2.6 Solution Techniques

Many algorithms have been proposed for solving the bilevel programming problem. Most of these algorithms have been designed for special cases of the problem. While some of the techniques used are suitable for only the form of the problem being considered by the authors, others use principles that are applicable to more general bilevel problems. The various algorithms can be classified[2], according to the solution approach used, into one of the following groups:

- Extreme point search or vertex enumeration algorithms;
- Descent direction algorithms;
- Linear complementarity algorithms;
- Branch and bound algorithms;
- Penalty function algorithms.

The basic motivation and structure of the algorithms in each class are described in this section.

This section is not meant to provide a complete review of all published algorithms for bilevel programming. Rather, it is intended as a brief introduction to existing approaches to the problem, so that our proposed algorithm can be viewed in context with other solution methods.

2.6.1 Extreme Point Search Algorithms

The family of algorithms referred to as extreme point search or vertex enumeration algorithms are used to solve linear bilevel problems. They are motivated by a theoretical result of Bialas and Karwan [21]. We shall consider the linear problem in the following form:

$$\text{LBP}: \quad \min_{x,y} \quad F(x,y) = a^T x + b^T y$$
$$\text{subject to} \quad y \in R(x) = \{y \in \mathbb{R}^m : y \text{ solves LLP}(x)\},$$

---

[2] The Alexandrov-Dennis [4] algorithm, noted on page 9 of this text, does not fit into just one category.

where

$$\mathrm{LlIP}(x): \min_{y} \quad f(y) = c^T y \quad \text{subject to} \quad A^T x + B^T y \geq d.$$

Bialas and Karwan showed that a solution of LBP must be a vertex, or extreme point, of the feasible region (in terms of x and y) of the lower level problem. Therefore, it must be a vertex of the simplex

$$\rho = \{(x, y) \in \mathbb{R}^{n+m} : A^T x + B^T y \geq d\}.$$

While a solution of LBP must be a vertex of ρ, not all vertices of ρ are feasible for LBP.

The goal of extreme point search algorithms is to examine all vertices (x, y) of ρ to identify a vertex which is feasible for LBP (for which y solves $\mathrm{LlIP}(x)$) and which minimizes $F(x, y)$ over all feasible points. However, as the region ρ may have an exponential number of vertices, simple enumeration of the vertices is too expensive. The various extreme point search algorithms use different techniques to efficiently search the vertices. Some of the proposed search techniques (for example, [31], [39], and [59]) examine only vertices which are feasible for LBP. However, the method we shall examine, proposed by Bialas and Karwan [21] and described due to its simplicity, iterates over infeasible points and terminates as soon as a feasible point is identified.

The algorithm begins by finding a global solution (x^1, y^1) to the reduced problem

$$\mathrm{LP}_1 : \min_{x,y} \quad F(x, y) \quad \text{subject to} \quad (x, y) \in \rho.$$

If (x^1, y^1) is not in the induced region, iteration k, for $k = 1, 2, \cdots$, proceeds in the following manner.

- Let V^k be the vertices visited in iterations $1, 2, \cdots, k-1$, and let W^k be the vertices of ρ which are neighbors of the points in V^k.

- Define the new point (x^{k+1}, y^{k+1}) by solving the linear problem

$$\text{LP}_k: \min_{x,y} \; F(x,y) \quad \text{subject to} \quad (x,y) \in W^k \setminus V^k.$$

- Update the iteration count $k = k + 1$.
- Repeat until y^k solves LLP(x^k).

This algorithm, which identifies the global solution of LBP, is called the Kth best algorithm because it terminates after K iterations, where (x^K, y^K) corresponds to the Kth lowest value of F over the simplex ρ.

The theory behind extreme point search algorithms cannot be extended to more general bilevel problems. Because nonlinear single level problems do not generally have solutions at vertices, the motivating theory is not applicable, in general, to nonlinear problems. However, Nicholls [58] has developed a grid search algorithm (based on the vertex enumeration technique) for a specific, two variable nonlinear bilevel problem. Because of the very specialized form of the problem, which was designed to represent the complete operations of an aluminum smelter, the author was able to divide the constraints into different groups (linear, convex quadratic, concave quadratic and those involving inverses of the two variables). Each group of constraints was handled individually. Currently, it is not known whether this technique can be extended to the more general forms of bilevel problems considered here.

2.6.2 Descent Direction Algorithms

Descent direction algorithms view the bilevel programming problem BP solely in terms of the upper level variables $x \in \mathbb{R}^n$. The lower level variables y are considered only in relation to the upper level variables. Since y is constrained to be a solution of LLP(x)

at a feasible point (x, y), y is an implicit function of x. It is assumed that the rational reaction set $R(x)$ contains a single entry for each feasible x. For convenience, we shall denote this solution $y^*(x)$ to emphasize the dependence of y on x.

Starting from a feasible point $x \in \mathbb{R}^n$, descent direction algorithms attempt to find a direction $w \in \mathbb{R}^n$ along which the objective function of the bilevel problem is decreased. A step $\alpha > 0$ is calculated along the descent direction and a new point $(x + \alpha w)$ is chosen which provides reasonable decrease in the objective and which is feasible for the bilevel problem. Feasibility requires calculating $y^*(x+\alpha w)$. The process continues until a descent direction cannot be defined and the current point is a local solution of the bilevel problem.

The various descent direction algorithms which have been proposed have the basic structure outlined above, but differ in the way that the descent direction is calculated. One major concern facing researchers using descent algorithms is the availability of the gradient of the objective function $\nabla_x F(x, y^*(x))$ at a feasible point. Using the chain rule for differentiation,

$$\nabla_x F(x, y^*(x)) = \nabla_x F(x, y) + \nabla_y F(x, y) \nabla_x y^*(x).$$

While $\nabla_x F(x, y)$ and $\nabla_y F(x, y)$ are assumed to exist, the gradient $\nabla_x y^*(x)$ may not exist, and even if it does, may be difficult to calculate. Assuming that the gradient exists, some descent direction algorithms (for example, the techniques cited in [52] and [53]) have concentrated on ways to approximate $\nabla_x y^*(x)$.

Savard and Gauvin [62] have proposed a different approach to the problem of defining a descent direction. Instead of calculating or estimating $\nabla_x y^*(x)$ and using this vector to define a descent direction, the authors propose calculating the actual steepest descent direction for the nonlinear bilevel problem. They describe a linear-quadratic bilevel problem (in which all functions are linear except for the lower level objective function which is quadratic) whose solution is the steepest descent direction for BP. The structure of the linear-quadratic bilevel problem can be exploited so that it is easier to solve than the nonlinear bilevel problem.

Once a descent direction w has been defined for the bilevel problem, regardless of the manner in which it was defined, a positive step size α must be determined such that an acceptable decrease is obtained in the objective. At the same time, the new point must be feasible. The new point $(x(\alpha), y(\alpha))$ must be calculated, where

$$\begin{aligned} x(\alpha) &= x + \alpha t \\ y(\alpha) &= y^*(x + \alpha t). \end{aligned}$$

Therefore, for each trial value of α considered, LLP($x(\alpha)$) must be solved. Solving these problems considerably raises the cost of performing the descent direction calculation in the upper level variable space.

These algorithms have been described for nonlinear problems. Numerical results for a few large-scale nonlinear bilevel problems have been quoted in [53], but the other descent direction techniques have cited results for only a few small problems.

2.6.3 Linear Complementarity Algorithms

Because bilevel problems are so difficult to solve, many techniques for solving them attempt to do so by solving a series of simpler, one level problems. Linear complementarity algorithms use this general approach to solve linear bilevel problems of the form

$$\begin{aligned} \text{LBP}_+ : \quad & \min_{x,y} \quad F(x,y) = a^T x + b^T y \\ & \text{subject to} \quad y \in R(x) = \{y \in I\!R^m : y \text{ solves LlIP}_+(x)\} \end{aligned}$$

where

$$\text{LlIP}_+(x): \ \min_y \ f(y) = c^T y \quad \text{subject to} \quad A^T x + B^T y \geq d, \ \ x, y \geq 0.$$

The representative algorithm described here is due to Júdice and Faustino [51], and is an improvement of the algorithm proposed by Bialas and Karwan [21].

The initial step in the Júdice-Faustino algorithm involves defining an equivalent one level problem by replacing the solution constraint connecting the upper and lower level problems with conditions that must hold at solutions of the primal and dual forms of the lower level problem.

$$\text{1-LBP}_+ : \quad \min_{x,y,\lambda,\alpha,\beta} \; F(x,y) \quad \text{subject to} \quad (x,y,\lambda,\alpha,\beta) \in \text{KKT} \;,$$

where

$$\begin{aligned} \text{KKT} \; = \; \{ & \\ & -d + A^T x + B^T y && = \alpha \\ & c - B\lambda && = \beta \\ & x^T \beta = \lambda^T \alpha && = 0 \\ & x, y, \lambda, \alpha, \beta && \geq 0 \\ \}. & \end{aligned}$$

Note that KKT includes points which satisfy the Karush-Kuhn-Tucker necessary optimality conditions for the lower level problem.

The vector λ, the Lagrange multipliers for the lower level problem, is the solution of the dual of the lower level problem. The vectors α and β are the slack variables for the primal and dual problems, respectively. Note that the complementary slackness conditions for the primal and dual problems are the third set of constraints listed in KKT. The problem 1-LBP_+ is not a linear problem. It is a nonconvex problem in which all but the complementary slackness conditions are linear.

Linear complementarity algorithms have been used to identify solutions of linear and quadratic problems by finding points which satisfy a set of conditions which must hold at a solution. To use linear complementarity techniques to solve 1-LBP$_+$ (and therefore to solve LBP$_+$), the authors introduce a variable ω and convert the objective function into a constraint

$$a^T x + b^T y \leq \omega.$$

The bilevel problem is now a problem in ω:

$$\mathrm{P}_\omega: \quad \min \quad \omega \quad \text{subject to } (x, y, \lambda, \alpha, \beta) \in \Upsilon(\omega),$$

where

$$\Upsilon(\omega) \;=\; \{(x, y, \lambda, \alpha, \beta) \in \text{ KKT } : w - a^T x - b^T y \geq 0\}.$$

Linear complementarity techniques are used by the authors to find a feasible point in $\Upsilon(\omega)$, for any value of ω.

If $\Upsilon(\omega)$ is empty, then ω is a lower bound on the value of $F(x, y)$ at a global solution of LBP$_+$. Increasing the value of ω and invoking a complementarity algorithm to find a feasible point of $\Upsilon(\omega)$, for the new value of ω, either finds a feasible point of LBP$_+$ or further increases the lower bound on the solution. If there is a feasible point in $\Upsilon(\omega)$ for the current value of ω, then ω is an upper bound on the optimal objective function value of LBP$_+$. By decreasing ω in this case, this upper bound can be reduced. Modifying ω in the manner described here will reduce the gap between the upper and lower bounds on the solution of LBP$_+$. This is the technique used by the authors to find an ϵ-global solution of LBP$_+$ (see [51] for an explanation of this term).

Unlike the extreme point search algorithms, the linear complementarity algorithms do not exploit the fact that the solution of the linear bilevel problem occurs at a vertex of the feasible region of the lower level problem. Combined with the fact that the Karush-Kuhn-Tucker (KKT) conditions (excluding the complementary slackness conditions) are linear for a quadratic problem, this means that this approach can also be used to solve bilevel problems whose upper level objective is linear and whose lower level problem is a convex quadratic problem.

2.6.4 Branch and Bound Algorithms

Like linear complementarity algorithms, branch and bound algorithms involve solving a series of "simpler" problems rather than solving the bilevel problem directly. Once again the KKT necessary conditions are used to define a one level optimization problem,

$$\begin{array}{llll} \mathrm{BP}_{KKT}: \min\limits_{x,y,\lambda} F(x,y) & \text{subject to} & G_i(x,y) \geq 0 & i \in T, \\ & \text{and} & \nabla_y \mathcal{L}(x,y,\lambda) = 0, & \\ & & g_i(x,y) \geq 0, & i \in P, \\ & & \lambda_i g_i(x,y) = 0, & i \in P, \\ & & \lambda_i \geq 0, & i \in P, \end{array}$$

where

$$\mathcal{L}(x,y,\lambda) = f(x,y) - \sum_{i=1}^{p} \lambda_i g_i(x,y)$$

is the Lagrangian function of the lower level problem defined at x, λ is the associated vector of Lagrange multipliers, and

$$\nabla_y \mathcal{L}(x,y,\lambda) = \left(\frac{\partial}{\partial y_i}\mathcal{L}(x,y,\lambda)\right)_{i\in M}$$

for $M = \{1,\cdots,m\}$, is its gradient with respect to the vector y. As mentioned previously, the complementary slackness conditions are generally nonlinear and nonconvex, and it follows that BP_{KKT} is as well.

Because the complementary slackness conditions are generally the most difficult constraints to satisfy in solving BP_{KKT}, branch and bound algorithms attempt to defer introducing those conditions into the solution process for as long as possible. This goal is achieved by building a tree of problems derived from BP_{KKT}. At the root, or initial node, of the tree is the problem

$$P_0: \min_{x,y,\lambda} \; F(x,y) \quad \text{subject to} \quad \begin{array}{ll} G_i(x,y) \geq 0, & i \in T, \\ \nabla_y \mathcal{L}(x,y,\lambda) = 0, & \\ g_i(x,y) \geq 0, & i \in P, \\ \lambda_i \geq 0, & i \in P, \end{array}$$

which is BP_{KKT} without the complementary slackness conditions. This problem is solved and the solution is used to construct the subtree in the following manner.

Assume that the problem at node k, denoted P_k, has been solved.

- If the complementary slackness condition $\lambda_j g_j(x,y) = 0$ is violated, then two children of node k are defined. One of the children, node k_1, contains problem P_k with the added constraint $\lambda_j = 0$. The other child, at node k_2, contains P_k plus the constraint $g_j(x,y) = 0$. Therefore, any solution of the problems at nodes k_1 and k_2 satisfies the jth complementary slackness condition.

- If there is no solution to the problem at node k, then the subtree rooted at node k is not expanded further because all problems in the subtree would be infeasible.

- If the solution to the problem at node k satisfies all the complementary slackness conditions, then a solution of BP_{KKT} has been identified. Its objective function value is compared to the best solution found so far. Because the solution just found is also a solution of the problems defined at the children of the current node, the tree is not expanded further. The subtree will not yield a better solution of BP_{KKT} than the current point.

The various branch and bound algorithms (as found in [3], [12], [13], [41], [43], [48], and [59]) use different search order rules for expanding the tree. Some use depth first search procedures and others use breadth first procedures. No individual approach seems better in general, although some may be more suited to different problems or solution goals.

Although the branch and bound technique is applicable to the nonlinear bilevel programming problem, most of the algorithms have been designed for linear or quadratic problems. Only [3] and [41] consider the nonlinear problem, and both of them place restrictions on the nonlinear bilevel problem being solved. Without these restrictions, the amount of work required to solve one nonlinear problem in the search tree may overwhelm the simplicity of the algorithm, and the technique may not be useful for a general nonlinear bilevel problem.

2.6.5 Penalty Function Algorithms

Many researchers have used penalty functions to define "simpler" problems in order to find a solution of BP. Unlike the other classes of algorithms described here, there is not a common general approach shared by all the penalty function algorithms.

Several techniques exist in which the lower level problem is replaced by a series of unconstrained penalty function problems. In [1], the authors use the solution of the penalty function subproblem to define the gradient $\nabla_x y^*(x)$ (see Section 2.6.2) to be used within a descent direction algorithm. In [2], the same authors instead replace the lower level problem with the stationarity condition of the unconstrained penalty function problem defined in their earlier work. In [50], the authors combine a weighted penalty function for the lower level problem with the upper level objective function to define a second penalty function which is minimized to find a solution of BP. These techniques have been designed to solve nonlinear problems, and numerical results are presented are for several small problems.

In [6], Anandalingham and White introduce, for the linear bilevel problem, a penalty function which consists of the upper level objective function and a weighted penalty term for the solution constraint. The penalty term is the duality gap between the the primal and dual versions of the lower level problem. By varying the weight and minimizing the penalty function, a solution of the linear bilevel problem can be found. The function is proven to be exact by showing that there exists a finite value of the penalty parameter for which some solution of the penalty function is also a solution of the linear bilevel problem. Marcotte and Zhu [57] propose a similar penalty function technique for the more general bilevel problem GBP described in Section 2.3. The constraint sets are assumed to be convex and some additional conditions are placed on the form of the problem. The new penalty function (which again is based on the duality gap of the lower level problem) is proven to be exact. Theoretical results are illustrated with two small examples, but a complete algorithm is not presented.

For the linear case, Bi, Calamai and Conn ([17] and [19]) convert the bilevel problem to a one level problem using the KKT necessary conditions, as done previously in defining BP_{KKT}. An exact ℓ_1 penalty function is defined based on the new one level problem. A minimum of the one level problem is found by modifying the penalty parameter and solving a series of unconstrained penalty function problems. Conditions are presented for which this minimum is also a minimum of the bilevel problem. Since these solutions may be local minima, the authors also include a globalization step based on objective function cuts to complete their algorithm. Numerical results are presented for linear problems.

In [55], Luo *et al.* describe a differentiable exact penalty function for the nonlinear bilevel problem whose lower level problem is a quadratic problem. While the technique seems promising, no numerical results are presented.

In [63] and [64], Scholtes *et al.* analyze and solve mathematical programs with equilibrium constraints. The authors present a series of optimality conditions for these problems, and present a trust region algorithm for solving them which uses exact, piecewise smooth penalty functions. The algorithm presented is theoretical and is incomplete with regard to the solution of the trust region subproblems.

2.7 A New Algorithm

In [20], Bi, Calamai, and Conn introduce the idea of extending their linear technique to nonlinear problems. In this thesis, this concept is fully developed. Subsequent chapters describe in detail the one level problem that is solved instead of the bilevel problem, along with the penalty function used, and the techniques used to solve the penalty function subproblems. Convergence and numerical results are presented.

Chapter 3

The Proposed Algorithm

3.1 Introduction

An algorithm is proposed which solves nonlinear bilevel problems by solving a "simpler" problem instead. The technique of Bi, Calamai and Conn for linear bilevel problems (as outlined in the previous chapter) is extended to the nonlinear case. The bilevel problem BP is transformed into a related, one level problem. A solution of the one level problem is obtained using an exact penalty function algorithm.

This chapter includes the derivation of the related problem, along with a description of some of the advantages and disadvantages of solving this problem instead of the original bilevel problem. The choice of a penalty function technique, in general, and the ℓ_1 penalty function, specifically, is motivated. A top level view of the penalty function algorithm for solving the related problem is then presented. The remainder of this chapter discusses the trust region framework used to solve instances of the penalty function subproblem.

3.2 On Solving BP

The bilevel problem BP is a very difficult problem to solve. Consider the process involved in identifying a feasible point (x, y):

1. Find $x \in \mathbb{R}^n$ such that the feasible region of the lower level problem

$$\rho(x) = \{y \in \mathbb{R}^m : g_i(x, y) \geq 0, \ i \in P\}$$

   is not empty.

2. Solve the lower level problem,

$$\text{LLP}(x) : \ \min_y \ f(x, y) \quad \text{subject to} \quad y \in \rho(x)$$

   to define the rational reaction set $R(x)$.

3. Ensure that x is feasible for BP by verifying that

$$\exists y \in R(x) \text{ such that } G_i(x, y) \geq 0, \ i \in T.$$

Minimizing the objective function $F(x, y)$ over all feasible (x, y) requires significantly more work.

Because of the difficulty in solving BP directly, we propose transforming it into a one level problem, and then solving the new problem using standard optimization techniques.

Initially, for presentation purposes, assume that LLP(x), for all feasible x, is a convex problem over $\mathbb{R}^m$, with a convex objective function $f(x, y)$ and concave constraint functions $g_i(x, y)$, for $i \in P$. With this assumption and under an appropriate constraint qualification[1], the KKT optimality conditions for the lower level problem at x are both

[1] For example, in this problem, Slater's condition requires that $\rho(x)$, the feasible region of LLP(x), is nonempty for all feasible x.

necessary and sufficient for identifying global solutions of LLP(x). In other words, $R(x)$, the set of solutions of LLP(x), is equivalent to $K(x)$, where

$$K(x) = \{y \in \mathbb{R}^m : \exists \lambda \in \mathbb{R}^p \text{ such that } (y, \lambda) \text{ satisfy the KKT optimality conditions K1-K4 for LLP}(x)\},$$

and

K1. A solution y of LLP(x) must be feasible: $g_i(x, y) \geq 0$, for $i \in P$.

K2. The Lagrange multipliers λ are nonnegative: $\lambda_i \geq 0$, for $i \in P$.

K3. The complementary slackness conditions are satisfied: $\lambda_i g_i(x, y) = 0$, for $i \in P$.

K4. The point y, with the multipliers λ, is a stationary point, with respect to y, of the Lagrangian function for LLP(x), that is, $c(x, y, \lambda) = 0$, where

$$\begin{aligned} c_i(x, y, \lambda) &= \frac{\partial}{\partial y_i}\mathcal{L}(x, y, \lambda) \\ &= \frac{\partial}{\partial y_i}\left(F(x, y) - \lambda^T g(x, y)\right) \\ &= \frac{\partial}{\partial y_i}F(x, y) - \sum_{i \in P} \lambda_i \frac{\partial}{\partial y_i} g_i(x, y) \\ &= 0, \end{aligned}$$

for $i \in M = \{1, \cdots, m\}$.

Therefore, under the stated assumptions, BP is equivalent to the problem

$$\begin{aligned} \text{BP}_{KKT}: \min_{x,y,\lambda} \quad F(x, y) \quad \text{subject to} \quad & G_i(x, y) \geq 0, & i \in T, \\ & g_i(x, y) \geq 0, & i \in P, \\ & \lambda_i \geq 0, & i \in P, \\ & \lambda_i g_i(x, y) = 0, & i \in P, \\ & c_i(x, y, \lambda) = 0, & i \in M. \end{aligned}$$

Note that the equivalence will not hold unless the optimal point satisfies a constraint qualification for the lower level problem. It is possible that a lower level problem will fail to satisfy a constraint qualification for all x.

Due to the relationship between BP and BP_{KKT}, we can solve BP by solving BP_{KKT} instead. Because BP_{KKT} is a one-level problem, traditional techniques for solving nonlinear problems can be applied to find a solution of BP.

To identify a feasible point of BP_{KKT} (and therefore of BP), we just need to find $(x, y, \lambda) \in \mathbb{R}^{n+m+p}$ which satisfies the constraints of BP_{KKT}. However, this apparent simplification may be misleading. While BP_{KKT} is a one level problem, it is still a difficult problem to solve. Consider that satisfying $c_i(x, y, \lambda) = 0$ involves identifying a stationary point of the Lagrangian function of LLP(x). Also, even if the upper level problem of BP is a convex problem, the complementary slackness conditions are generally nonconvex and nonlinear. Therefore, BP_{KKT} is usually a nonconvex problem. Finally, note that BP_{KKT} is defined over $n + m + p$ variables and has $t + m + 3p$ constraints, which may be a significant increase over the number of variables and constraints in BP.

We now remove the assumption that LLP(x) is convex, and examine how the absence of this assumption affects the relationship between BP and BP_{KKT}. As stated previously, conditions K1, K2, K3, and K4 identify a global solution of the convex problem LLP(x). Without the assumption of convexity, these conditions must still be satisfied at any local solution of LLP(x). However, if a value (y, λ) is identified which satisfies these conditions, the point may be either a local minimizer, maximizer or saddle point of LLP(x). Therefore, the conditions are necessary at a solution of LLP(x), but are not sufficient to identify a solution of LLP(x). Problems BP and BP_{KKT} are not generally equivalent without the convexity assumption.

While not equivalent, BP and BP_{KKT} are still closely related. Any solution of BP must be a solution of BP_{KKT}, and all feasible points of BP are feasible points of BP_{KKT}. While the converse is not true, some solutions of BP_{KKT} will be solutions of BP, and some feasible points of BP_{KKT} will be feasible points of BP. However, a feasible point of BP_{KKT} for which (y, λ) is a saddle or maximum point of LLP(x) is not feasible for BP. The following result summarizes the relationship between BP and BP_{KKT}.

Observation 3.1 *The following relationships hold between BP and BP_{KKT}:*

- *If LLP(x) is convex over $\mathbb{R}^m$, for all feasible $x \in \mathbb{R}^n$, then BP and BP_{KKT} are equivalent.*
- *If LLP(x) is not convex over $\mathbb{R}^m$ for some feasible $x \in \mathbb{R}^n$, then*
  - *$\mathcal{I}$, the induced region of BP, is contained in $\mathcal{S}$, the feasible region of BP_{KKT}, that is, $\mathcal{I} \subseteq \mathcal{S}$.*
  - *The solution set of BP is contained in the solution set of BP_{KKT}.*
  - *If, for some feasible $x \in \mathbb{R}^n$, LLP(x) has saddle points or local maxima, then the feasible region of BP_{KKT} contains points which are not feasible for BP, that is, $\mathcal{S} \not\subseteq \mathcal{I}$.*
  - *The solution set of BP_{KKT} is not necessarily contained in the solution set of BP.*

Due to the strong relationship between BP and BP_{KKT}, we solve BP by solving BP_{KKT} instead. However, it is understood that solutions of BP_{KKT} may be found which are not feasible for BP.

As mentioned above, BP_{KKT} is a nonlinear, nonconvex problem, which may itself be difficult to solve, and which may have many more variables and constraints than BP. In order to reduce the number of constraints in the one level problem, we note the following equivalence.

Lemma 3.2 *The set*

$$\{(x, y, \lambda) \in \mathbb{R}^{n+m+p} : \ K1,\ K2, \text{ and } \ K3 \text{ are satisfied}\}$$

is equivalent to the set

$$\{(x, y, \lambda) \in \mathbb{R}^{n+m+p} : \min(g_i(x, y), \lambda_i) = 0, \ \text{for } i \in P\}.$$

Proof: The result follows immediately. □

Using the relationship in Lemma 3.2, BP_{KKT} can be written equivalently, but more compactly, as a problem with $n + m + p$ variables and $t + m + p$ constraints.

$$\begin{aligned} \mathrm{BP}_C: \ \min_{x,y,\lambda} \ F(x, y) \quad \text{subject to} \quad & G_i(x, y) \geq 0, && i \in T, \\ & \min(g_i(x, y), \lambda_i) = 0, && i \in P, \\ & c_i(x, y, \lambda) = 0, && i \in M. \end{aligned}$$

If, for some $i \in P$, $g_i(x, y) = \lambda_i$, the min function is nondifferentiable at (x, y, λ). The loss of differentiability is a direct result of reducing the number of constraints in BP_{KKT} using the equivalence stated in Lemma 3.2.

3.3 Solving the Transformed Problem

As mentioned above, BP_C is a difficult problem to solve. In the most general case, it is nonlinear, nonconvex and nondifferentiable, and finding a feasible point is nontrivial. When solving BP_C, the proposed algorithm avoids the difficulty of finding an initial feasible point by allowing the iterates to be infeasible. The algorithm attempts to minimize the objective function and move towards feasibility at the same time. A penalty function framework, based on the ideas in Conn [36], is used to achieve this goal.

A penalty function for BP_C combines the objective function $F(x, y)$ with a weighted, nonnegative penalty for infeasibility,

$$r_\mu(x, y, \lambda) = F(x, y) + \mu\nu(x, y, \lambda),$$

where $\mu > 0$ is called the penalty parameter, and $\nu(x, y, \lambda)$ is the unscaled, nonnegative, penalty associated with the current point. The parameter μ is used to balance the possibly conflicting goals of minimizing $F(x, y)$ and obtaining feasibility (which is attained by reducing $\nu(x, y, \lambda)$ to zero).

In the ℓ_1 penalty function for BP_C, each constraint contributes one term to the penalty $\nu(x, y, \lambda)$, as detailed in Table 3.1. The penalty terms are simply the magnitude of the

Constraint		Penalty Term
$i \in T$,	$G_i(x, y) \geq 0$	$-\min(G_i(x, y), 0)$
$i \in P$,	$\min(g_i(x, y), \lambda_i) = 0$	$\lvert\min(g_i(x, y), \lambda_i)\rvert$
$i \in M$,	$c_i(x, y, \lambda) = 0$	$\lvert c_i(x, y, \lambda)\rvert$.

Table 3.1: Composition of the ℓ_1 Penalty Term

violation of the constraint. The larger the violation becomes, the larger the penalty term becomes. If any constraint is satisfied, then the corresponding penalty term is zero.

These penalty terms may be nondifferentiable at some points (x, y, λ). If, for $i \in T$, $G_i(x, y) = 0$ (i.e. an upper level constraint is exactly satisfied), then the corresponding penalty term is nondifferentiable with respect to (x, y, λ). For the remaining constraints, if the argument of the absolute value function is zero, then the penalty term is again nondifferentiable. Note that this type of nondifferentiability is in addition to the non-differentiability inherent in the constraints for $i \in P$. However, the nondifferentiabilities introduced by the ℓ_1 penalty terms are similar in nature to the constraint nondifferentiabilities. This similarity is a major reason for the use of the ℓ_1 penalty function to solve BP_C. The traditional techniques for handling the ℓ_1 nondifferentiabilities (see, for example, [14]) can be generalized to handle the constraint nondifferentiabilities.

Let $\nu_1(x, y, \lambda)$ be the sum of all the ℓ_1 penalty terms,

$$\nu_1(x, y, \lambda) = -\sum_{i \in T} \min(G_i(x, y), 0) + \sum_{i \in P} |\min(g_i(x, y), \lambda_i)| + \sum_{i \in T} |c_i(x, y, \lambda)|,$$

and let $p_\mu(x, y, \lambda)$ be the ℓ_1 penalty function

$$p_\mu(x, y, \lambda) = F(x, y) + \mu\nu_1(x, y, \lambda).$$

Consider the problem

$$\mathrm{PF}(\mu): \min_{x,y,\lambda} \; p_\mu(x, y, \lambda),$$

for different values of μ.

- If μ is relatively small compared with F/ν_1, it may be possible to obtain unbounded descent in p_μ because a decrease in $F(x, y)$ dominates the infeasibility penalty. Similarly, a local minimizer of p_μ may be infeasible for BP_C. Assuming a solution of BP_C exists, increasing the value of μ decreases the possibility that p_μ is unbounded below or that a local minimizer exists which is infeasible for BP_C.

- If μ is relatively large compared with F/ν_1, decreasing the value of $F(x, y)$ is not as desirable when minimizing p_μ as decreasing the penalty terms. Therefore, the observed decrease in $F(x, y)$ may be very slow in relation to the rate at which a feasible point is approached. This may result in slow convergence of the algorithm to a solution of BP_C, even when close to feasibility.

Also note that as μ becomes increasingly larger, the implementation of an algorithm to minimize p_μ may begin to experience numerical difficulties. Small changes in the penalty terms are magnified within p_μ and small changes in $F(x, y)$ become numerically insignificant, resulting in a loss of precision.

Ideally, the value of μ balances the goal of minimizing $F(x, y)$ with the goal of attaining feasibility.

Results are now presented which motivate the penalty function algorithm. The theorem presents a property of the differentiable problem BP_{KKT}. The corollary extends the property to BP_C, the nondifferentiable problem being solved.

Both results make a nondegeneracy assumption. Some additional terminology is required to fully define the meaning of nondegeneracy in this context, so the definition is delayed until Section 3.5.2.

Theorem 3.3 *Assuming that BP_{KKT} is a nondegenerate problem, there exists a finite value of $\mu^* > 0$ such that, for $\mu > \mu^*$, there is a local minimizer of*

$$\begin{aligned} q_\mu(x, y, \lambda) &= F(x, y) - \mu \sum_{i \in T} \min(0, G_i(x, y)) - \mu \sum_{i \in P} \min(0, g_i(x, y)) \\ &\quad - \mu \sum_{i \in P} \min(0, \lambda_i) + \mu \sum_{i \in P} |\lambda_i g_i(x, y)| + \mu \sum_{i \in M} |c_i(x, y, \lambda)|, \end{aligned}$$

denoted $(x^q_\mu, y^q_\mu, \lambda^q_\mu)$, which is also a local minimizer of BP_{KKT}.

Proof: This is a restatement of Theorem 1 in Pietrzkowski[60]. Note that q_μ is the ℓ_1 penalty function defined from BP_{KKT} in the same way that p_μ is defined from BP_C, with one penalty term for each constraint. □

Corollary 3.4 *Assuming that BP_C is a nondegenerate problem, there exists a finite value $\mu^* > 0$, such that for $\mu > \mu^*$, there is a local minimizer of p_μ, denoted $(x_\mu, y_\mu, \lambda_\mu)$, which is also a local minimizer of BP_C.*

Proof: This result follows immediately from the above theorem, the equivalence of BP_{KKT} and BP_C, and the resulting equivalence of q_μ and p_μ. □

This result must be interpreted correctly. First, note that the critical value of the penalty parameter, μ^*, is a finite value. Therefore, the ℓ_1 penalty function is an exact penalty function. Unfortunately, the actual value of μ^* depends on values at a solution of BP_C, and are not known *a priori*. However, even if μ^* is known explicitly, solving PF(μ^*) may not always identify a local solution of BP_C. While there exists a local solution of p_{μ^*} which is also a local minimizer of BP_C, the penalty function may have other local solutions (which may or may not be solutions of BP_C) or may even be unbounded.

Despite the limitations expressed above, Corollary 3.4 provides a very important result. Solving PF(μ), for $\mu > \mu^*$, may provide a solution of BP_C, and therefore, may provide a solution of BP. However, since μ^* is unknown, this suggests solving several PF(μ) problems, for an increasing sequence of values of μ, until a solution of BP_C is identified. This proposal is presented more formally in the next section.

3.4 Penalty Function Algorithm

Algorithm 3.1 (Penalty Function Framework)

1. *Set $\mu_{\max}$, an upper bound on μ.*
2. *Choose initial values (x^0, y^0, λ^0) and $\mu^0 \in (0, \mu_{\max}]$. Set $k = 0$.*
3. *If $\mu^k > \mu_{\max}$, then stop without identifying a solution of BP_C.*
4. *Solve PF(μ), for $\mu = \mu^k$, starting from (x^k, y^k, λ^k).*
5. *Evaluate the solution $(x^{k+1}, y^{k+1}, \lambda^{k+1})$:*

(a) *If $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is feasible for BP_C, then stop with a solution of BP_C,*

$$(x_\mu, y_\mu, \lambda_\mu) = (x^{k+1}, y^{k+1}, \lambda^{k+1}).$$

(b) *If $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is infeasible for BP_C, then update $\mu^{k+1} > \mu^k$ and set $k = k + 1$. Repeat from Step 3.*

(c) *If PF(μ) is unbounded, then $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is the last point reached before the unboundedness was detected. Update $\mu^{k+1} > \mu^k$, and set $k = k+1$. Repeat from Step 3.*

Note the use of $\mu_{\max}$ to prevent μ from becoming too large in the solution process. The value of $\mu_{\max}$ is implementation dependent.

The following outcomes of Algorithm 3.1 are possible:

O1. A local solution of PF(μ^k) is feasible for BP_C.

O2. A local solution of PF(μ^{k-1}) is infeasible for BP_C and $\mu^k > \mu_{\max}$.

O3. The problem PF(μ^{k-1}) is unbounded below and $\mu^k > \mu_{\max}$.

It is possible that the feasible region of BP_C is empty, or that BP_C is unbounded below. These possibilities can account for the final two outcomes listed above. However, it is also possible that BP_C has feasible, bounded, local solutions, but the algorithm terminates with either outcome O2 or O3. The performance of the algorithm is influenced by the initial choices (x^0, y^0, λ^0) and μ^0. Different outcomes may be observed for different starting values. For this reason, whenever either of the final two outcomes are observed, new choices of (x^0, y^0, λ^0) and μ^0 are made, and the entire process is repeated from Step 2 of the algorithm.

If the algorithm terminates with outcome O1, then $(x_\mu, y_\mu, \lambda_\mu)$ is a local solution of BP_C. If (y_μ, λ_μ) is a local solution of LLP(x_μ), then a local solution of BP has been identified. Otherwise, (y_μ, λ_μ) is a saddle or maximum point of LLP(x_μ), and the point (x_μ, y_μ, z_μ) is not a solution of BP. In this case, a new starting point and penalty parameter value can be chosen, and the process repeated from Step 2.

3.5 Solutions of the Penalty Function Subproblems

Before describing the proposed algorithm for solving PF(μ), for a fixed value of μ, the structure of the penalty function is investigated and necessary conditions at a local minimum of p_μ are developed. These conditions identify possible solutions of PF(μ), and will be used to develop termination criteria for the PF(μ) algorithm. They are also used constructively in the algorithm, as described in the next chapter.

The following definitions are required before proceeding further.

Definition 3.1

1. *Let $w \in \mathbb{R}^{n+m+p}$ represent the vector $(x^T, y^T, \lambda^T)^T$.*

2. *The variable λ_i can be written in terms of w as follows:*

$$\lambda_i(w) = w_{n+m+i} = w^T e_{n+m+i},$$

where e_{n+m+i}, for $i \in P$, is the $(n+m+i)$th column of the identity matrix in $\mathbb{R}^{n+m+p}$.

3. *For any $w \in \mathbb{R}^{n+m+p}$,*

$$T^0(w) \quad = \quad \{i \in T : G_i(w) = 0\}$$

$$\begin{aligned}
T'(w) &= T \setminus T^0(w) \\
P_\lambda(w) &= \{i \in P : \lambda_i(w) < g_i(w)\} \\
P_\lambda^0(w) &= \{i \in P_\lambda(w) : \lambda_i(w) = 0\} \\
P_\lambda'(w) &= P_\lambda(w) \setminus P_\lambda^0(w) \\
P_g(w) &= \{i \in P : g_i(w) < \lambda_i(w)\} \\
P_g^0(w) &= \{i \in P_g(w) : g_i(w) = 0\} \\
P_g'(w) &= P_g(w) \setminus P_g^0(w) \\
P_=(w) &= \{i \in P : \lambda_i(w) = g_i(w)\} \\
P_=^0(w) &= \{i \in P_=(w) : \lambda_i(w) = g_i(w) = 0\} \\
P_='(w) &= P_=(w) \setminus P_=^0(w) \\
M^0(w) &= \{i \in M : c_i(w) = 0\} \\
M'(w) &= M \setminus M^0(w).
\end{aligned}$$

4. *For any $v \in \mathbb{R}$,*

$$sign[v] = \left\{ \begin{array}{cl} 1 & \text{if } v > 0 \\ -1 & \text{if } v < 0 \\ 0 & \text{otherwise} \end{array} \right\} \text{ and } neg[v] = \left\{ \begin{array}{cl} 1 & \text{if } v < 0 \\ 0 & \text{otherwise} \end{array} \right\}.$$

Also, define $pos[v] = neg[-v]$.

3.5.1 Examining the Structure of the Penalty Function

The following result is used to simplify the value of $p_\mu(w + \alpha d)$ for an arbitrary direction $d \in \mathbb{R}^{n+m+p}$.

Lemma 3.5 *For any $d \in \mathbb{R}^{n+m+p}$, there exists a value $\alpha_1 > 0$ such that the following conditions are all satisfied for $\alpha \in [0, \alpha_1]$:*

- *For* $i \in T'(w)$, $neg[G_i(w + \alpha d)] = neg[G_i(w)]$.
- *For* $i \in P_\lambda(w)$, $\lambda_i(w + \alpha d) < g_i(w + \alpha d)$.
- *For* $i \in P'_\lambda(w)$, $sign[\lambda_i(w + \alpha d)] = sign[\lambda_i(w)]$.
- *For* $i \in P_g(w)$, $g_i(w + \alpha d) < \lambda_i(w + \alpha d)$.
- *For* $i \in P'_g(w)$, $sign[g_i(w + \alpha d)] = sign[g_i(w)]$.
- *For* $i \in P'_=(w)$, $sign[\min(\lambda_i(w + \alpha d), g_i(w + \alpha d))] = sign[\lambda_i(w)]$.
- *For* $i \in M'(w)$, $sign[c_i(w + \alpha d)] = sign[c_i(w)]$.

Proof: We consider the first case in detail.

For $i \in T'(w)$, consider the function $G_i(w) \neq 0$. The continuity of G_i ensures that either there exists $\hat{\alpha}_i > 0$ such that

$$\text{neg}[G_i(w + \alpha d)] = \text{neg}[G_i(w)] \text{ for } 0 \leq \alpha < \hat{\alpha}_i, \text{ and } G_i(w + \hat{\alpha}_i d) = 0,$$

or

$$\text{neg}[G_i(w + \alpha d)] = \text{neg}[G_i(w)] \text{ for all } \alpha > 0.$$

Let $\alpha_i^G = \hat{\alpha}_i$ in the first situation, and $\alpha_i^G = \infty$, in the second. The value of α_1 must satisfy $\alpha_1 < \alpha_i^G$ for all $i \in T'(w)$. Because α_i^G is nonzero, an acceptable value of α_1 always exists.

By analyzing the remaining functions

$$\begin{array}{lll} & g_i(w + \alpha d) - \lambda_i(w + \alpha d) & \text{for } i \in P_\lambda(w) \cup P_g(w), \\ & \lambda_i(w + \alpha d) & \text{for } i \in P'_\lambda(w) \cup P'_=(w) \\ & g_i(w + \alpha d) & \text{for } i \in P'_g(w) \cup P'_=(w) \\ \text{and} & c_i(w + \alpha d) & \text{for } i \in M'(w), \end{array}$$

in a similar manner, further conditions are placed on α_1.

Any α_1 which satisfies the required conditions is sufficient for the result to hold. □

The following simplification is made throughout the rest of the thesis.

Notation 3.1 *For convenience, the argument w is omitted from the penalty term index sets. Unless explicitly stated otherwise, the point w is the intended argument.*

Using the above result, we can separate the function $p_\mu(w+\alpha d)$ into two parts based on the differentiability of penalty terms at $\alpha = 0$.

Lemma 3.6 *For an arbitrary direction $d \in \mathbb{R}^{n+m+p}$, and $\alpha \in [0, \alpha_1]$,*

$$p_\mu(w+\alpha d) = \delta(w+\alpha d) + \mu\eta(w+\alpha d),$$

where $\delta(w+\alpha d)$ is a differentiable function over the interval $\alpha \in [0, \alpha_1]$, and $\eta(w+\alpha d)$ is not differentiable at $\alpha = 0$.

Proof: Using the result in Lemma 3.5, we can write, for $\alpha \in [0, \alpha_1]$,

$$\begin{aligned} p_\mu(w+\alpha d) &= F(w+\alpha d) - \mu \sum_{i\in T'} \min(G_i(w+\alpha d), 0) \\ &\quad -\mu \sum_{i\in T^0} \min(G_i(w+\alpha d), 0) + \mu \sum_{i\in P_\lambda} |\lambda_i(w+\alpha d)| \\ &\quad +\mu \sum_{i\in P_g} |g_i(w+\alpha d)| + \mu \sum_{i\in P_=} |\min(\lambda_i(w+\alpha d), g_i(w+\alpha d))| \\ &\quad +\mu \sum_{i\in M'} |c_i(w+\alpha d)| + \mu \sum_{i\in M^0} |c_i(w+\alpha d)|. \end{aligned}$$

For $i \in P'_=$, note that

$$\min(\lambda_i(w+\alpha d), g_i(w+\alpha d)) = \lambda_i(w+\alpha d) + \min\left(0, g_i(w+\alpha d) - \lambda_i(w+\alpha d)\right).$$

Now, continuing with $\alpha \in [0, \alpha_1]$,

$$p_\mu(w+\alpha d) = \delta(w+\alpha d) + \mu\eta(w+\alpha d),$$

where

$$\begin{aligned}\delta(w+\alpha d) &= F(w+\alpha d) - \mu \sum_{i\in T'} \text{neg}[G_i(w)]G_i(w+\alpha d)\\ &+\mu \sum_{i\in P'_\lambda} \text{sign}[\lambda_i(w)]\lambda_i(w+\alpha d) + \mu \sum_{i\in P'_g} \text{sign}[g_i(w)]g_i(w+\alpha d)\\ &+\mu \sum_{i\in P'_=} \text{sign}[\lambda_i(w)]\lambda_i(w+\alpha d) + \mu \sum_{i\in M'} \text{sign}[c_i(w)]c_i(w+\alpha d)\end{aligned}$$

is a continuous, differentiable function over the interval $\alpha \in [0, \alpha_1]$, and

$$\begin{aligned}\eta(w+\alpha d) &= -\sum_{i\in T^0} \min(G_i(w+\alpha d), 0) + \sum_{i\in P^0_\lambda} |\lambda_i(w+\alpha d)| + \sum_{i\in P^0_g} |g_i(w+\alpha d)|\\ &+ \sum_{i\in P'_=} \text{sign}[\lambda_i(w)] \min(0, g_i(w+\alpha d) - \lambda_i(w+\alpha d))\\ &+ \sum_{i\in P^0_=} |\min(\lambda_i(w+\alpha d), g_i(w+\alpha d))| + \sum_{i\in M^0} |c_i(w+\alpha d)|.\end{aligned}$$

While $\eta(w+\alpha d)$ is nondifferentiable at $\alpha = 0$, some of the terms may be differentiable for $\alpha > 0$, depending on how their values change along the direction d. □

The two functions δ and η will be used to obtain further information about the penalty function along $w + \alpha d$.

3.5.2 First Order Necessary Optimality Conditions

The following definitions are required in the proof of the next result.

Definition 3.2

1. *The vector* $\gamma(w) = \nabla_w \delta(w)$,

$$\begin{aligned}\gamma(w) &= \nabla F(w) - \mu \sum_{i\in T'} neg[G_i(w)]\nabla G_i(w) + \mu \sum_{i\in P'_\lambda} sign[\lambda_i(w)]\nabla \lambda_i(w)\\ &+\mu \sum_{i\in P'_g} sign[g_i(w)]\nabla g_i(w) + \mu \sum_{i\in P'_=} sign[\lambda_i(w)]\nabla \lambda_i(w)\\ &+\mu \sum_{i\in M'} sign[c_i(w)]\nabla c_i(w),\end{aligned}$$

is called the gradient of the differentiable part, or simply the gradient, of the penalty function at w. The matrix $B(w) = \nabla_w^2 \delta(w)$ is the Hessian of the differentiable part of p_μ.

2. *The function η is called the nondifferentiable part of the penalty function.*

3. *The activity matrix $\mathcal{A}(w)$ consists of the gradients of the defining terms in the function η at w, that is, $\mathcal{A}(w)$ contains of the following columns.*

$$\begin{array}{lll} & \nabla \lambda_i(w), & \text{for } i \in P_\lambda^0 \cup P_=^0 \\ & \nabla g_i(w), & \text{for } i \in P_g^0 \cup P_=^0 \\ & \nabla g_i(w) - \nabla \lambda_i(w), & \text{for } i \in P_=' \\ & \nabla G_i(w), & \text{for } i \in T^0 \\ \text{and} & \nabla c_i(w), & \text{for } i \in M^0. \end{array}$$

4. *If the columns of $\mathcal{A}(w)$ are linearly independent, then the point w is a nondegenerate or regular point. If all points are nondegenerate points, then the penalty function subproblem is a nondegenerate problem.*

5. *The columns of the matrix $Z(w)$ form an orthogonal basis for the space orthogonal to the space spanned by the gradients of the activities, that is,*

$$\mathcal{A}(w)^T Z(w) = 0 \text{ and } Z(w)^T Z(w) = I.$$

6. *If $Z(w)$ and $\gamma(w)$ satisfy $Z(w)^T \gamma(w) = 0$, then w is a stationary point for the penalty function p_μ.*

7. *For any differentiable function $h : \mathbb{R}^{n+m+p} \to \mathbb{R}$, point $w \in \mathbb{R}^{n+m+p}$, direction $d \in \mathbb{R}^{n+m+p}$ and step size $\alpha > 0$, the Generalized Rayleigh Quotient is given by*

$$\Omega(h, w, \alpha d) = \frac{2}{\alpha^2 \|d\|_2^2} \left(h(w + \alpha d) - h(w) - \alpha d^T \nabla h(w) \right).$$

This value is a measure of the curvature of the function h at w for positive steps α along the direction d.

Again, for convenience, the following notational simplification is introduced.

Notation 3.2 *The argument of the vector $\gamma(w)$ and the matrices $B(w)$, $A(w)$ and $Z(w)$ will be omitted. Unless stated otherwise, the current point w is the intended argument.*

We next establish a set of conditions which must be satisfied by a minimum point of the penalty function.

Lemma 3.7 *If w is not a stationary point of p_μ and p_μ is bounded, then for $d_s = -ZZ^T\gamma$, there exists $\alpha_s > 0$ such that $p_\mu(w + \alpha d_s) < p_\mu(w)$, for $0 < \alpha < \alpha_s$.*

Proof: The differentiable and nondifferentiable parts of $p_\mu(w + \alpha d)$ will be considered separately.

We first examine, in greater detail, the differentiable part of p_μ, for $0 \leq \alpha \leq \alpha_1$.

$$\begin{aligned} \delta(w + \alpha d_s) &= \delta(w) + \alpha d_s^T \nabla \delta(w) + \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(\delta, w, \alpha d_s) \\ &= p_\mu(w) - \alpha \gamma^T Z Z^T \gamma + \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(\delta, w, \alpha d_s) \\ &= p_\mu(w) - \alpha \|Z^T \gamma\|_2^2 + \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(\delta, w, \alpha d_s). \end{aligned}$$

Next, consider the nondifferentiable part of p_μ. Each set of terms in η can be considered separately. For $i \in T^0$,

$$\begin{aligned} G_i(w + \alpha d_s) &= G_i(w) + \alpha d_s^T \nabla G_i(w) + \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(G_i, w, \alpha d_s) \\ &= \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(G_i, w, \alpha d_s), \end{aligned}$$

because $G_i(w) = 0$ (from $i \in T^0$) and $d_s^T \nabla G_i(w) = 0$ (since $d_s^T A = -\gamma^T Z Z^T A = 0$). It similarly follows that

$$\begin{array}{llcl} \text{for } i \in P_\lambda^0 \cup P_=^0, & \lambda_i(w + \alpha d_s) & = & 0 \\ \text{for } i \in P_g^0 \cup P_=^0, & g_i(w + \alpha d_s) & = & \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(g_i, w, \alpha d_s) \\ \text{for } i \in P_=', & g_i(w + \alpha d_s) - \lambda_i(w + \alpha d_s) & = & \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(g_i - \lambda_i, w, \alpha d_s) \\ \text{for } i \in M^0, & c_i(w + \alpha d_s) & = & \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega(c_i, w, \alpha d_s). \end{array}$$

Therefore, for $0 \le \alpha \le \alpha_1$, we can write

$$\eta(w + \alpha d_s) = \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega_\eta(w, \alpha d_s),$$

where we define, for the nondifferentiable function η,

$$\begin{aligned} \Omega_\eta(w, \alpha d_s) \;=\; & -\sum_{i \in T^0} \min(\Omega(G_i, w, \alpha d_s), 0) + \sum_{i \in P_g^0} |\Omega(g_i, w, \alpha d_s)| \\ & + \sum_{i \in P_='} \operatorname{sign}[\lambda_i(w)] \min(0, \Omega(g_i - \lambda_i, w, \alpha d_s)) \\ & + \sum_{i \in P_=^0} |\min(0, \Omega(g_i, w, \alpha d_s))| + \sum_{i \in M^0} |\Omega(c_i, w, \alpha d_s)|. \end{aligned}$$

Let $\Omega_{p_\mu}(w, \alpha d_s) = \Omega(\delta, w, \alpha d_s) + \Omega_\eta(w, \alpha d_s)$. Since p_μ is assumed to be bounded, Ω_{p_μ} is bounded as well. Now, it follows from Lemma 3.6, that, for $0 \le \alpha \le \alpha_1$,

$$p_\mu(w + \alpha d_s) = p_\mu(w) - \alpha \|Z^T \gamma\|_2^2 + \frac{1}{2}\alpha^2 \|d_s\|_2^2 \Omega_{p_\mu}(w, \alpha d_s).$$

There exists $\alpha_s \in (0, \alpha_1)$, such that, with respect to α above, the first order term dominates the second order term for $\alpha < \alpha_s$. Since the first order term is negative, it follows that $p_\mu(w + \alpha d_s) < p_\mu(w)$ for $0 < \alpha < \alpha_s$. □

Corollary 3.8 *If w is a minimum point of p_μ, then w is a stationary point of p_μ.*

Proof: Follows immediately from the above lemma, since descent in p_μ is not possible from a minimum point w. □

Definition 3.3

1. *At a stationary point of* p_μ, *the vector* Ψ *satisfying* $\mathcal{A}\Psi = \gamma$ *is called the vector of Lagrange multipliers. These multipliers will be referenced as illustrated in Table 3.2.*

Multiplier		*Corresponding Column of* $\mathcal{A}$
Ψ_i^G	$i \in T^0$	$\nabla G_i(w)$
Ψ_i^λ	$i \in P_\lambda^0 \cup P_=^0$	$\nabla \lambda_i(w)$
Ψ_i^g	$i \in P_g^0 \cup P_=^0$	$\nabla g_i(w)$
$\Psi_i^{g-\lambda}$	$i \in P_=^\prime$	$\nabla g_i(w) - \nabla \lambda_i(w)$
Ψ_i^c	$i \in M^0$	$\nabla c_i(w)$.

Table 3.2: Notation for Lagrange Multipliers

2. *Let* $e_i^G \in \mathbb{R}^{n+m+p}$, *for* $i \in T^0$, *refer to the identity column corresponding to the location of* $\nabla G_i(w)$ *in* $\mathcal{A}$. *Similarly define the identity columns* $e_i^\lambda \in \mathbb{R}^{n+m+p}$ *(for* $i \in P_\lambda^0 \cup P_=^0$*),* $e_i^g \in \mathbb{R}^{n+m+p}$ *(*$i \in P_g^0 \cup P_=^0$*),* $e_i^{g-\lambda} \in \mathbb{R}^{n+m+p}$ *(for* $i \in P_=^\prime$*), and* $e_i^c \in \mathbb{R}^{n+m+p}$ *(for* $i \in M^0$*).*

3. *The set* $P_=^\prime$ *at* w *is further refined.*

$$\begin{aligned} P_=^+ &= \{i \in P_=^\prime : \lambda_i(w) > 0\} \\ P_=^- &= \{i \in P_=^\prime : \lambda_i(w) < 0\} = P_=^\prime \setminus P_=^+. \end{aligned}$$

4. *A stationary point w of p_μ is called a first order point of p_μ if the following conditions are all satisfied.*

$$\begin{array}{rcccll} 0 & \le & \Psi_i^G & \le & \mu, & \text{for } i \in T^0, \\ -\mu & \le & \Psi_i^\lambda & \le & \mu, & \text{for } i \in P_\lambda^0, \\ -\mu & \le & \Psi_i^g & \le & \mu, & \text{for } i \in P_g^0, \\ 0 & \le & \Psi_i^{g-\lambda} & \le & \mu, & \text{for } i \in P_=^-, \\ 0 & \le & \Psi_i^\lambda, \Psi_i^g & \le & \mu, & \text{for } i \in P_=^0, \\ & & \Psi_i^\lambda + \Psi_i^g & \le & \mu, & \text{for } i \in P_=^0, \\ -\mu & \le & \Psi_i^c & \le & \mu, & \text{for } i \in M^0, \\ & & P_=^+ & = & \emptyset. & \end{array}$$

5. *A multiplier Ψ_i is said to be in kilter if it falls within the corresponding range required of a first order point.*

Lemma 3.9 *If w is a nondegenerate stationary point of p_μ but is not a first order point and p_μ is bounded, then there exists a constant $\alpha_D > 0$ and a direction d_D which satisfies at least one of the following conditions*

- *if* $\exists j \in T^0 : \Psi_j^G \notin [0,\mu]$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^G] e_j^G$,
- *if* $\exists j \in P_\lambda^0 : \Psi_j^\lambda \notin [-\mu,\mu]$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^\lambda] e_j^\lambda$,
- *if* $\exists j \in P_g^0 : \Psi_j^g \notin [-\mu,\mu]$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^g] e_j^g$,
- *if* $\exists j \in P_=^+ : \Psi_j^{g-\lambda} > -\mu$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^{g-\lambda} + \mu] e_j^{g-\lambda}$,
- *if* $\exists j \in P_=^+ : \Psi_j^{g-\lambda} < 0$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^{g-\lambda}] e_j^{g-\lambda}$,
- *if* $\exists j \in P_=^- : \Psi_j^{g-\lambda} \notin [0,\mu]$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^{g-\lambda}] e_j^{g-\lambda}$,
- *if* $\exists j \in P_=^0 : \Psi_j^\lambda \notin [0,\mu]$, *then* d_D *satisfies* $A^T d_D = -sign[\Psi_j^\lambda] e_j^\lambda$,

- *if* $\exists j \in P^0_= : \Psi^g_j \notin [0,\mu]$, *then* d_D *satisfies* $\mathcal{A}^T d_D = -sign[\Psi^g_j]e^g_j$,
- *if* $\exists j \in P^0_= : \Psi^\lambda_j + \Psi^g_j > \mu$, *then* d_D *satisfies* $\mathcal{A}^T d_D = -sign[\Psi^\lambda_j + \Psi^g_j - \mu](e^\lambda_j + e^g_j)$,
- *if* $\exists j \in M^0 : \Psi^c_j \notin [-\mu,\mu]$, *then* d_D *satisfies* $\mathcal{A}^T d_D = -sign[\Psi^c_j]e^c_j$,

such that $p_\mu(w+\alpha d_D) < p_\mu(w)$ *for* $0 < \alpha < \alpha_D$.

Proof: Note that the assumption of nondegeneracy ensures the existence of at least one such dropping direction d_D. In all cases in which a single activity is being dropped, we can write that $\mathcal{A}^T d_D = \sigma_j e_j$ for some $\sigma_j = \pm 1$ and some cardinal unit vector $e_j \in \mathbb{R}^{n+m+p}$.

We consider, in detail, the first case listed above, namely that G_j is being dropped from the active set for some $j \in T^0$. For $0 \le \alpha \le \alpha_1$, from Lemma 3.5, we have that

$$\begin{aligned}\delta(w+\alpha d_D) &= p_\mu(w) + \alpha d_D^T \gamma + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega(\delta, w, \alpha d_D) \\ &= p_\mu(w) + \alpha d_D^T \mathcal{A}\Psi + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega(\delta, w, \alpha d_D) \\ &= p_\mu(w) - \alpha|\Psi^G_j| + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega(\delta, w, \alpha d_D),\end{aligned}$$

since $d_D^T \mathcal{A}\Psi = -\text{sign}[\Psi^G_j]\Psi^G_j$. The terms in η must be examined individually. Note that

$$\begin{aligned}G_j(w+\alpha d_D) &= G_j(w) + \alpha d^T \nabla G_j(w) + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega(G_j, w, \alpha d_D) \\ &= -\alpha \text{sign}[\Psi^G_j] + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega(G_j, w, \alpha d_D).\end{aligned}$$

Since p_μ is bounded, the Ω terms are bounded as well. Therefore, there exists $\alpha_2 \in (0,\alpha_1)$ such that for $0 \le \alpha \le \alpha_2$, the first order term dominates the value of G_j,

$$\text{neg}[G_j(w+\alpha d_D)] = \text{neg}[-\text{sign}[\Psi^G_j]] = \text{neg}[-\Psi^G_j].$$

For $i \in T^0 \setminus \{j\}$, $d_D^T \nabla G_i(w) = 0$ by the definition of d_D. Similarly, first order change in the other activities is zero along d_D.

Therefore, for $0 \leq \alpha \leq \alpha_2$,

$$\begin{aligned}\eta(w+\alpha d_D) &= -\alpha \min(-\text{sign}[\Psi_j^G], 0) + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega_\eta(w, \alpha d_D) \\ &= \alpha(1 - \text{neg}[\Psi_j^G]) + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega_\eta(w, \alpha d_D),\end{aligned}$$

where

$$\begin{aligned}\Omega_\eta(w, \alpha d_D) &= -\text{neg}[G_j(w+\alpha d_D)]\Omega(G_j, w, \alpha d_D) \\ &\quad - \sum_{i \in T^0 \setminus \{j\}} \min(\Omega(G_i, w, \alpha d_D), 0) + \sum_{i \in P_g^0} |\Omega(g_i, w, \alpha d_D)| \\ &\quad + \sum_{i \in P_=^{\prime}} \text{sign}[\lambda_i(w)] \min(0, \Omega(g_i - \lambda_i, w, \alpha d_D)) \\ &\quad + \sum_{i \in P_=^0} |\min(0, \Omega(g_i, w, \alpha d_D))| + \sum_{i \in M^0} |\Omega(c_i, w, \alpha d_D)|.\end{aligned}$$

Using Lemma 3.6, for $0 \leq \alpha \leq \alpha_2$,

$$p_\mu(w+\alpha d_D) = p_\mu(w) + \alpha(-|\Psi_j^G| + \mu - \mu\text{neg}[\Psi_j^G]) + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega_{p_\mu}(w, \alpha d_D),$$

where $\Omega_{p_\mu}(w, \alpha d_D) = \Omega(\delta, w, \alpha d_D) + \mu\Omega_\eta(w, \alpha d_D)$.

Consider, if $\Psi_j^G < 0$, then

$$p_\mu(w+\alpha d_D) = p_\mu(w) + \alpha\Psi_j^G + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega_{p_\mu}(w, \alpha d_D).$$

Otherwise, for $\Psi_j^G > \mu$,

$$p_\mu(w+\alpha d_D) = p_\mu(w) + \alpha(\mu - \Psi_j^G) + \frac{1}{2}\alpha^2 \|d_D\|_2^2 \Omega_{p_\mu}(w, \alpha d_D).$$

In either case, the first order term is negative, and there exists $\alpha_D \in (0, \alpha_2)$, such that, for $0 \leq \alpha < \alpha_D$, the first order term dominates in $p_\mu(w+\alpha d_D)$. Therefore, for $0 < \alpha < \alpha_D$, $p_\mu(w+\alpha d_D) < p_\mu(w)$.

Dropping directions d_D which satisfy any of the other conditions listed above are shown to be descent directions for p_μ from w in a similar manner. The details are omitted. □

Corollary 3.10 *If w is a nondegenerate minimum point of p_μ, then w is a first order point of p_μ.*

Proof: Follows immediately from the above lemma since descent in p_μ is not possible from a minimum point w. □

3.5.3 Second Order Necessary Optimality Conditions

Definition 3.4

1. *The matrix H, defined at w,*

$$\begin{aligned} H &= B - \sum_{i \in T^0} \Psi_i^G \nabla^2 G_i(w) - \sum_{i \in P_g^0} \Psi_i^g \nabla^2 g_i(w) - \sum_{i \in P_=^!} \Psi_i^{g-\lambda} \nabla^2 g_i(w) \\ &\quad - \sum_{i \in P_=^0} \Psi_i^g \nabla^2 g_i(w) - \sum_{i \in M^0} \Psi_i^c \nabla^2 c_i(w), \end{aligned}$$

   is called the Hessian of the Lagrangian function at w.

2. *If w is a first order point of the penalty function and $Z^T H Z$ is positive semidefinite, then w is a second order point of p_μ.*

The material presented here is greatly influenced by the work of Coleman and Conn [35] in developing the necessary second order optimality conditions for the ℓ_1 penalty function for nonlinear programming.

Let w^0 be a minimum point of p_μ. To establish a set of second order conditions that must be satisfied at w^0, a new, differentiable nonlinear problem P_0 is defined. The objective function $q_0(w)$ is defined using the activity sets at w^0.

$$q_0(w) = F(w) - \mu \sum_{i \in T^-} G_i(w) + \mu \sum_{i \in P_\lambda^+} \lambda_i(w) - \mu \sum_{i \in P_\lambda^-} \lambda_i(w) + \mu \sum_{i \in P_g^+} g_i(w)$$

$$-\mu \sum_{i\in P_g^-} g_i(w) + \mu \sum_{i\in P_=^+} \lambda_i(w) - \mu \sum_{i\in P_=^-} \lambda_i(w)$$
$$+\mu \sum_{i\in M^+} c_i(w) - \mu \sum_{i\in M^-} c_i(w), \tag{3.1}$$

where

$$\begin{aligned}
T^- &= \{i \in T' : G_i(w^0) < 0\} \\
T^+ &= \{i \in T' : G_i(w^0) > 0\} \\
P_\lambda^+ &= \{i \in P_\lambda' : \lambda_i(w^0) > 0\} \\
P_\lambda^- &= \{i \in P_\lambda' : \lambda_i(w^0) < 0\} \\
P_g^+ &= \{i \in P_g' : g_i(w^0) > 0\} \\
P_g^- &= \{i \in P_g' : g_i(w^0) < 0\} \\
P_=^+ &= \{i \in P_=' : \lambda_i(w^0) = g_i(w^0) > 0\} \\
P_=^- &= \{i \in P_=' : \lambda_i(w^0) = g_i(w^0) < 0\} \\
M^+ &= \{i \in M' : c_i(w^0) > 0\} \\
M^- &= \{i \in M' : c_i(w^0) < 0\}.
\end{aligned}$$

Next, we define a set of constraints, also using the activity sets corresponding to w^0. Let

$$\begin{aligned}
S_G = \{w \in \mathbb{R}^{n+m+p} :\ & G_i(w) \geq 0 && \text{for } i \in T^+ \cup T^0 \\
& G_i(w) \leq 0 && \text{for } i \in T^-\}, \\
S_\lambda = \{w \in \mathbb{R}^{n+m+p} :\ & \lambda_i(w) \geq 0 && \text{for } i \in P_\lambda^+ \\
& \lambda_i(w) = 0 && \text{for } i \in P_\lambda^0 \\
& \lambda_i(w) \leq 0 && \text{for } i \in P_\lambda^- \\
\text{and }\ & \lambda_i(w) \leq g_i(w) && \text{for } i \in P_\lambda\},
\end{aligned}$$

$$
\begin{aligned}
S_g = \{w \in \mathbb{R}^{n+m+p}: \ & g_i(w) \geq 0 && \text{for } i \in P_g^+ \\
& g_i(w) = 0 && \text{for } i \in P_g^0 \\
& g_i(w) \leq 0 && \text{for } i \in P_g^- \\
\text{and } \ & g_i(w) \leq \lambda_i(w) && \text{for } i \in P_g\}, \\
S_= = \{w \in \mathbb{R}^{n+m+p}: \ & \lambda_i(w) \geq 0 && \text{for } i \in P_=^+ \\
& \lambda_i(w) = g_i(w) = 0 && \text{for } i \in P_=^0 \\
& g_i(w) = 0 && \text{for } i \in P_=^0 \\
& \lambda_i(w) \leq 0 && \text{for } i \in P_=^- \\
\text{and } \ & g_i(w) \geq \lambda_i(w) && \text{for } i \in P_=^+ \cup P_=^-\}, \\
S_c = \{w \in \mathbb{R}^{n+m+p}: \ & c_i(w) \geq 0 && \text{for } i \in M^+ \\
& c_i(w) = 0 && \text{for } i \in M^0 \\
& c_i(w) \leq 0 && \text{for } i \in M_-\},
\end{aligned}
$$

and

$$S_0 = S_G \cup S_\lambda \cup S_g \cup S_= \cup S_c.$$

Lemma 3.11 *There exists $N(w^0)$, a small neighborhood of w^0, in which the sets S_0 and S_0' are equivalent, where*

$$
\begin{aligned}
S_0' = \{ \ & w \in \mathbb{R}^{n+m+p}: \\
& G_i(w) \geq 0, && \textit{for } i \in T^0, \\
& \lambda_i(w) = 0, && \textit{for } i \in P_\lambda^0 \cup P_=^0, \\
& g_i(w) = 0, && \textit{for } i \in P_g^0 \cup P_=^0, \\
& g_i(w) - \lambda_i(w) \geq 0, && \textit{for } i \in P_=^+ \cup P_=^-, \\
& c_i(w) = 0, && \textit{for } i \in M^0\}.
\end{aligned}
$$

Proof: Follows from the continuity of the functions and their values at w^0. □

Next, consider the following differentiable nonlinear problem P_0 and its relationship to the penalty function.

$$P_0 : \qquad \min_{w} \; q_0(w) \text{ s.t. } w \in S_0'.$$

Lemma 3.12 *The point w^0 is a local minimizer of p_μ if and only if it is also a local minimizer of P_0.*

Proof: For $w \in S_0$ and therefore for $w \in S_0 \cap N(w^0)$, $q_0(w) = p_\mu(w)$. From Lemma 3.11, it also follows that, for $w \in S_0' \cap N(w^0)$, $q_0(w) = p_\mu(w)$. If there exists a feasible point with a smaller value of q_0 within $S_0' \cap N(w^0)$, then this point also gives a lower value of p_μ. Similarly, if there exists a point with a lower value of p_μ in this region, then it also gives a lower value of q_0. Therefore, the local minima of the two problems in this region must coincide. □

The next result follows from the equivalence established above.

Corollary 3.13 *If w^0 is a nondegenerate local minimum of p_μ, then*

1. *the necessary first order conditions are satisfied, and*
2. *for all $d \in \mathcal{N}_0$*

$$\begin{aligned}
\mathcal{N}_0 = \; & \{d \in \mathbb{R}^{n+m+p} : \\
& d^T \nabla G_i(w^0) = 0 && \text{for } i \in T^0 \\
& d^T \nabla \lambda_i(w^0) = 0 && \text{for } i \in P_\lambda^0 \cup P_=^0 \\
& d^T \nabla g_i(w^0) = 0 && \text{for } i \in P_g^0 \cup P_=^0 \\
& d^T (\nabla g_i(w^0) - \nabla \lambda_i(w^0)) = 0 && \text{for } i \in P_=' \\
& d^T \nabla c_i(w^0) = 0 && \text{for } i \in M^0\}
\end{aligned}$$

it follows that

$$d^T H(w^0) d \geq 0,$$

where

$$\begin{aligned} H(w^0) &= B(w^0) - \sum_{i \in T^0} \Psi_i^G \nabla^2 G_i(w^0) - \sum_{i \in P_g^0} \Psi_i^g \nabla^2 g_i(w^0) - \sum_{i \in P_=^0} \Psi_i^g \nabla^2 g_i(w^0) \\ &\quad - \sum_{i \in P_=^\prime} \Psi_i^{g-\lambda} \nabla^2 g_i(w^0) - \sum_{i \in M^0} \Psi_i^c \nabla^2 c_i(w^0). \end{aligned}$$

Proof: The first result was established in Corollary 3.10 and the second result follows from the necessary optimality conditions for the nonlinear problem P_0. Note that the definition of $H(w^0)$ differs from Definition 3.4 in the use of $P_=^-$ rather than $P_=^\prime$. This change is possible because $P_=^\prime = P_=^+ \cup P_=^-$. The set $P_=^+$ is empty since w^0 satisfies the first order necessary conditions. □

Therefore, if w^0 is a nondegenerate local minimum of p_μ, it is a second order point of p_μ since

- $\mathcal{A}(w^0)\Psi = \gamma$ with the multipliers Ψ in kilter, and
- $Z^T H(w^0) Z$ is positive semidefinite.

A stricter set of second order necessary optimality conditions is established by examining the value of $p_\mu(w^0)$ to determine when a second order descent direction can be defined.

Lemma 3.14 *Let w^0 be a nondegenerate first order point of the penalty function. If there exists a direction z in the reduced space which is a direction of negative curvature for the matrix $Z^T \hat{B}(z) Z$, then $d = Zz$ is a descent direction for p_μ, where*

$$\begin{aligned} \hat{B}(z) &= B - \mu \sum_{i \in T^0} \theta_i \nabla^2 G_i(w^0) + \mu \sum_{i \in P_g^0} \vartheta_i \nabla^2 g_i(w^0) - \mu \sum_{i \in P_=^-} \theta_i \nabla^2 g_i(w^0) \\ &\quad - \mu \sum_{i \in P_=^0} \theta_i \nabla^2 g_i(w^0) + \mu \sum_{i \in M^0} \vartheta_i \nabla^2 c_i(w^0) \end{aligned}$$

and

- *for* $i \in T^0$, $\theta_i = neg[d^T \nabla^2 G_i(w^0)d]$
- *for* $i \in P_g^0$, $\vartheta_i = sign[d^T \nabla^2 g_i(w^0)d]$,
- *for* $i \in P_=^-$, $\theta_i = neg[d^T \nabla^2 g_i(w^0)d]$
- *for* $i \in P_=^0$, $\theta_i = neg[d^T \nabla^2 g_i(w^0)d]$
- *for* $i \in M^0$, $\vartheta_i = sign[d^T \nabla^2 c_i(w^0)d]$.

Proof: Assume that α is small enough so that the second order terms dominate the higher order terms. The differentiable and nondifferentiable parts of the penalty function at $w^0 + \alpha d$ can be expressed as follows.

$$
\begin{aligned}
\delta(w^0 + \alpha d) &= \delta(w^0) + \alpha d^T \gamma + \frac{1}{2}\alpha^2 d^T B d + o(\|\alpha\|^3) \\
&= p_\mu(w^0) + \frac{1}{2}\alpha^2 d^T B d + o(\|\alpha\|^3) \\
\eta(w^0 + \alpha d) &= \frac{1}{2}\alpha^2 \Bigg(- \sum_{i \in T^0} \min(0, d^T \nabla^2 G_i(w^0)d) + \sum_{i \in P_g^0} |d^T \nabla^2 g_i(w^0)d| \\
&\quad - \sum_{i \in P_=^-} \min(0, d^T \nabla^2 g_i(w^0)d) - \sum_{i \in P_=^0} \min(0, d^T \nabla^2 g_i(w^0)d) \\
&\quad + \sum_{i \in M^0} |d^T \nabla^2 c_i(w^0)d| \Bigg) + o(\|\alpha\|^3) \\
&= \frac{1}{2}\alpha^2 \Bigg(- \sum_{i \in T^0} \theta_i d^T \nabla^2 G_i(w^0)d + \sum_{i \in P_g^0} \vartheta_i d^T \nabla^2 g_i(w^0)d \\
&\quad - \sum_{i \in P_=^-} \theta_i d^T \nabla^2 g_i(w^0)d - \sum_{i \in P_=^0} \theta_i d^T \nabla^2 g_i(w^0)d \\
&\quad + \sum_{i \in M^0} \vartheta_i d^T \nabla^2 c_i(w^0)d \Bigg) + o(\|\alpha\|^3).
\end{aligned}
$$

Therefore,

$$\begin{aligned} p_\mu(w^0 + \alpha d) &= \delta(w^0 + \alpha d) + \mu\eta(w^0 + \alpha d) \\ &= p_\mu(w^0) + \frac{1}{2}\alpha^2 d^T \hat{B}(z) d + o(\|\alpha\|^3), \end{aligned}$$

where

$$\begin{aligned} \hat{B}(z) &= H + \sum_{i \in T^0} (\Psi_i^G - \mu\theta_i)\nabla^2 G_i(w^0) + \sum_{i \in P_g^0} (\Psi_i^g + \mu\vartheta_i)\nabla^2 g_i(w^0) \\ &\quad + \sum_{i \in P_=^-} (\Psi_i^{g-\lambda} - \mu\theta_i)\nabla^2 g_i(w^0) + \sum_{i \in P_=^0} (\Psi_i^g - \mu\theta_i)\nabla^2 g_i(w^0) \\ &\quad + \sum_{i \in M^0} (\Psi_i^c + \mu\vartheta_i)\nabla^2 c_i(w^0). \end{aligned}$$

Note that the two definitions of $\hat{B}(z)$ are equivalent. A direction of negative curvature for $Z^T \hat{B}(z) Z$ provides descent in p_μ. □

Corollary 3.15 *If w^0 is a nondegenerate local minimum of p_μ, then the matrix $Z^T \hat{B}(z) Z$ is positive semidefinite.*

The matrix $\hat{B}(z)$ defined above can have more than one value, since it is defined using the θ_i and ϑ_i values which depend on z. Note that each θ_i and ϑ_i can have two values, where, for simplicity we depart from the definition of sign[0] on page 41 and let sign[0] $= 1$. Therefore, in theory, there are an exponential (though finite) number of values of $\hat{B}$ possible. It is also possible that some of these combinations of θ and ϑ have no meaning. For example, if $\nabla^2 G_i(w^0) = 0$ for some $i \in T^0$, then the value $\theta_i = 1$ will never be meaningful in the context in which it is used here.

The second order optimality conditions proven so far are summarized below.

1. Two sets of second order conditions which must be satisfied at a local minimum of p_μ have been proven. The first one, that $Z^T H Z$ is positive semidefinite, is easily

verified. The second one, that $Z^T\hat{B}Z$ is positive semidefinite for each meaningful matrix $\hat{B}$ is more difficult to establish. We can check that each $Z^T\hat{B}Z$ is positive semidefinite, but it may be more difficult to determine if a particular $\hat{B}$ makes sense in terms of the given situation.

2. We have established a condition under which a descent direction can be defined from w^0, a nondegenerate first order point of the penalty function which is not a second order point. If we have identified a meaningful $\hat{B}$ for which $Z^T\hat{B}Z$ is indefinite, a direction of negative curvature can be calculated. However, that direction must correspond to the correct values of θ and ϑ to guarantee descent in the penalty function.

The following relationships illustrate that the two sets of optimality conditions are not generally equivalent.

Lemma 3.16 *If $Z^T H Z$ is positive semidefinite, then $Z^T\hat{B}Z$ is also positive semidefinite for all meaningful $\hat{B}$.*

Proof: Assume that $Z^T H Z$ is positive semidefinite for any direction z in the reduced space. Let $d = Zz$. Now, using the existing definitions of θ_i and ϑ_i, consider

- for $i \in T^0$, recall that $0 \le \Psi_i^G \le \mu$,
  - if $\theta_i = 0$, then $\Psi_i^G \ge 0$ and $d^T\nabla^2 G_i d \ge 0$,
  - if $\theta_i = 1$, then $\Psi_i^G - \mu\theta_i = \Psi_i^G - \mu \le 0$ and $d^T\nabla^2 G_i d < 0$.

  In either case, $(\Psi_i^G - \mu\theta_i)d^T\nabla^2 G_i d \ge 0$.
- for $i \in P_g^0$, recall that $-\mu \le \Psi_i^g \le \mu$,

 - if $\vartheta_i = -1$, then $\Psi_i^g + \mu\vartheta_i = \Psi_i^g - \mu \leq 0$ and $d^T\nabla^2 g_i d \leq 0$,
 - if $\vartheta_i = 1$, then $\Psi_i^g + \mu\vartheta_i = \Psi_i^g + \mu \geq 0$ and $d^T\nabla^2 g_i d \geq 0$.

 In either case, $(\Psi_i^g + \mu\vartheta_i)d^T\nabla^2 g_i d \geq 0$.

- for $i \in P_=^-$, as for $i \in T^0$, it follows that $(\Psi_i^{g-\lambda} - \mu\theta_i)d^T\nabla^2 g_i d \geq 0$.
- for $i \in P_=^0$, as for $i \in T^0$, it follows that $(\Psi_i^g - \mu\theta_i)d^T\nabla^2 g_i d \geq 0$.
- for $i \in M^0$, as for $i \in P_g^0$, it follows that $(\Psi_i^c + \mu\vartheta_i)d^T\nabla^2 c_i d \geq 0$.

The sum of all these terms must be nonnegative, i.e. $\hat{B}(z)$ is positive semidefinite. □

A corresponding result equating the definiteness for all meaningful $Z^T\hat{B}(z)Z$ with the definiteness of Z^THZ cannot be established. However, a result relating the definiteness of all possible values of $Z^T\hat{B}Z$ to that of Z^THZ can be proven. Let $\mathcal{B}$ be any possible value of $\hat{B}$,

$$\begin{aligned}\mathcal{B} &= H + \sum_{i\in T^0}(\Psi_i^G - \mu\theta_i)\nabla^2 G_i + \sum_{i\in P_g^0}(\Psi_i^g + \mu\vartheta_i)\nabla^2 g_i + \sum_{i\in P_=^-}(\Psi_i^{g-\lambda} - \mu\theta_i)\nabla^2 g_i \\ &\quad + \sum_{i\in P_=^0}(\Psi_i^g - \mu\theta_i)\nabla^2 g_i + \sum_{i\in M^0}(\Psi_i^c + \mu\vartheta_i)\nabla^2 c_i\end{aligned}$$

where each $\theta_i \in \{0,1\}$ and each $\vartheta_i = \pm 1$. Note that there are 2^K possible values for $\mathcal{B}$, where

$$K = |T^0| + |P_g^0| + |P_=^0| + |P_=^-| + |M^0|.$$

The set of meaningful values of $\hat{B}$ is a subset of the set of possible values of $\mathcal{B}$.

Lemma 3.17 *If $Z^T\mathcal{B}Z$ is positive semidefinite for all 2^K combinations of θ_i and ϑ_i, then Z^THZ is positive semidefinite.*

Proof: For any value of $\mathcal{B}$ and direction $d = Zz$, it follows that

$$\begin{aligned} d^T H d \geq & \sum_{i \in T^0} (\theta_i \mu - \Psi_i^G) d^T \nabla^2 G_i d - \sum_{i \in P_g^0} (\vartheta_i \mu + \Psi_i^g) d^T \nabla^2 g_i d + \sum_{i \in P_=^0} (\theta_i \mu - \Psi_i^g) d^T \nabla^2 g_i d \\ & + \sum_{i \in P_=^-} (\theta_i \mu - \Psi_i^{g-\lambda}) d^T \nabla^2 g_i d - \sum_{i \in M^0} (\vartheta_i \mu + \Psi_i^c) d^T \nabla^2 c_i d. \end{aligned}$$

In particular, it must be true for the following choice of θ_i and ϑ_i:

- for $i \in T^0$, $\theta_i = \text{pos}[d^T \nabla^2 G_i d]$,
- for $i \in P_g^0$, $\vartheta_i = -\text{sign}[d^T \nabla^2 g_i d]$,
- for $i \in P_=^0$, $\theta_i = \text{pos}[d^T \nabla^2 g_i d]$,
- for $i \in P_=^-$, $\theta_i = \text{pos}[d^T \nabla^2 g_i d]$,
- for $i \in M^0$, $\vartheta_i = -\text{sign}[d^T \nabla^2 c_i d]$.

For this choice of θ and ϑ it is easy to determine that the right hand side above must be nonnegative, i.e. that $d^T H d \geq 0$. Thus, we have proven that $Z^T H Z$ is positive semidefinite. □

The requirement that $Z^T \mathcal{B} Z$ is positive semidefinite for all values of $\mathcal{B}$ is significantly stronger than the requirement that $Z^T \hat{B} Z$ is positive semidefinite for all meaningful values of $\hat{B}$. For the latter, the definiteness is only over those z in the reduced space which correspond to meaningful values of $\hat{B}$. Meanwhile, the former requires the definiteness over all z in the reduced space for each $\mathcal{B}$. However, the condition on $\mathcal{B}$ is easier to check than the condition on $\hat{B}$.

If $Z^T H Z$ is indefinite, and z, a direction of negative curvature for $Z^T H Z$, is also a direction of negative curvature for $Z^T \hat{B}(z) Z$, then $d = Zz$ is a descent direction for p_μ from w^0. Such a condition is easily checked. If z does not correspond to a direction of negative curvature for $Z^T\hat{B}(z)Z$, then the algorithm may not be able to find a descent direction for p_μ. Problems of this nature, as described by Coleman and Conn in [33] and [35], are associated with penalty functions and other methods for solving nonlinear optimization problems.

3.6 Motivating Theory for Solving the Subproblem

Because of the complicated structure of the penalty function, it is preferable to deal with a simplified version of the function whenever possible. This simplified, or model, function should be easier to minimize than the penalty function. The model function should also be a good approximation of the penalty function, in a region about the current point, so that a direction which decreases the model function will also decrease the penalty function.

The idea just described is the basic motivation of trust region algorithms. While originally used in unconstrained optimization, trust region algorithms (for example, see [40]) are being used increasingly for constrained optimization, and have been shown to have good convergence results (both in theory and practice) for nonconvex problems. The performance of these algorithms on nonconvex problems is the primary reason that a trust region algorithm is being proposed to solve PF(μ).

Solving PF(μ) within a trust region framework involves modeling the change in the penalty function from the current point w along a direction d, using a "simpler" function $\phi(d)$. By restricting the norm of direction d when analyzing $\phi(d)$, the model function can be minimized over a region in which the model function is believed to be a good estimate of $p_\mu(w+d)$. The trust region subproblem, centered at point w, is therefore

$$\text{TR}(w,\Delta): \ \min_d \ \ \phi(d) \quad \text{subject to} \quad \|d\| \le \Delta,$$

where Δ is the current trust region radius.

A solution of TR(w, Δ), denoted d_T, is then used to evaluate the performance of the model. The actual decrease in the penalty function along d_T,

$$\chi_a(d_T) = p_\mu(w) - p_\mu(w + d_T),$$

is compared to the decrease predicted by the model,

$$\chi_p(d_T) = \phi(0) - \phi(d_T).$$

- If the actual and predicted decreases are in very close agreement, then the model function is assumed to provide a very good approximation of the penalty function over the current trust region. Therefore, the trial point $w + d_T$ is accepted as the new current point. Also, the performance of the model function over the current region indicates that the model function may be a good approximation over a larger region. The trust region radius at the new point is therefore increased from its current value.

- If the actual and predicted decreases are in reasonably close agreement, then the model function is assumed to provide a reasonable approximation of the penalty function over the current trust region. Therefore, while $w + d_T$ is accepted as the new current point, the trust region radius at the new point is kept at its current value. The performance of the model, while acceptable, did not indicate that the model function would provide a good approximation for the penalty function over a larger region.

- If the actual and predicted decreases are in poor agreement, then the model function does not provide a good approximation of the penalty function over the current trust region. The trial point $w + d_T$ is rejected, and the current point is maintained. Due to the poor performance of the model, the trust region radius is reduced from its current value. Because the model function should be a better estimate of the penalty function over a smaller region, the trust region subproblem is solved again from the current point with a reduced value of Δ. This process is repeated until the model provides an acceptable approximation of the penalty function.

By its definition, direction d_T provides descent in the model function. Therefore, $\chi_p(d_T) > 0$ is always satisfied. Note that if the direction d_T provides no decrease in the penalty function (that is, the actual decrease value is negative), the final situation listed above will be satisfied. Therefore, when the step d_T is accepted, it always decreases the penalty function.

3.7 Modeling the Penalty Function

For an arbitrary direction d, the model function ϕ should be "simpler" than the penalty function, while reflecting, as much as possible, the value in the penalty function along d.

Definition 3.5 *The model function ϕ is defined by replacing each component function of the penalty function by its quadratic Taylor's expansion approximation, and by retaining the penalty term structure of p_μ.*

$$\begin{aligned}
\phi(d) &= F(w) + \alpha d^T \nabla F(w) + \frac{1}{2} d^T \nabla^2 F(w) d \\
&\quad -\mu \sum_{i \in T} \min(G_i(w) + d^T \nabla G_i(w) + \frac{1}{2} d^T \nabla^2 G_i(w) d, 0) \\
&\quad +\mu \sum_{i \in P} |\min(\lambda_i(w) + d^T \nabla \lambda_i(w), g_i(w) + d^T \nabla G_i(w) + \frac{1}{2} d^T \nabla^2 g_i(w) d)| \\
&\quad +\mu \sum_{i \in M} |c_i(w) + d^T \nabla c_i(w) + \frac{1}{2} d^T \nabla^2 c_i(w) d|.
\end{aligned}$$

Note that ϕ is a piecewise quadratic function, which models the shape of the penalty function up to second order changes.

This choice of model function has the desirable property that $\phi(0) = p_\mu(w)$. Also, the gradient of the differentiable part of ϕ at zero is γ, the gradient of the differentiable part of p_μ at w. Similarly, the Hessian of the differentiable part of $\phi(0)$ is B, the Hessian of the differentiable part of p_μ at w. Like the penalty function, the model function has points of nondifferentiability.

To simplify the process used to solve the trust region subproblems TR(w, Δ), the ℓ_∞ norm is used in the distance constraint. With this choice of norm, the constraint, $\|d\|_\infty \leq \Delta$, is equivalent to placing simple bounds on the components of the descent direction, that is, $-\Delta \leq d_i \leq \Delta$, for $i = 1 : n + m + p$. Consequently, solving TR(w, Δ), for fixed values of w and Δ, involves minimizing a piecewise quadratic function over simple bounds.

3.8 Terminating the Algorithm

The trust region algorithm for solving PF(μ) can be terminated in one of two ways:

1. A decision that the current point w is a possible solution of PF(μ).

2. A decision that the problem PF(μ) is unbounded.

As proven in Corollary 3.13, a local minimum of the penalty function must be a second order point of p_μ. Therefore, when the current point satisfies all the conditions of a second order point, the algorithm concludes that it is a possible solution of the problem, and terminates. If w is feasible for the one level problem BP$_C$, as defined on page 34, then Algorithm 3.1 is also terminated. Otherwise, the penalty parameter μ is increased and a new penalty function subproblem is solved.

Detecting the unboundedness of PF(μ) within a trust region framework is somewhat more complicated than within a step length based algorithm. In the latter, if a step of unbounded length continues to decrease the objective function, the problem is unbounded. However, in the trust region framework, the length of the step at each iteration is limited by the distance constraint of the subproblem TR(w, Δ). In addition, an upper bound $\Delta_{\max}$ is often placed on the size of the trust region radius to avoid numerical difficulties associated with a very large value of Δ.

Within the proposed algorithm, a heuristic algorithm is used to determine unboundedness. It is concluded that the current penalty function subproblem PF(μ) is unbounded below if both of the following conditions are observed.

1. A sequence of $b_{\max}$ consecutive, very successful iterations are observed, for the algorithm parameter $b_{\max}$.
2. Over this sequence of iterations, the contribution of the penalty terms within the penalty function,

$$(p_\mu(w) - F(w))/\mu$$

   was not decreased.

The conditions above detect a sequence of iterations over which the penalty function is decreased significantly without the penalty terms being decreased. This suggests that decreases in the function F are outweighing the penalty for infeasibility, and that the penalty function is becoming unbounded. In this case, the trust region algorithm should return a result of unboundedness, along with the last point w encountered before the sequence of iterations described above.

3.9 Trust Region Algorithm for the Subproblem

Note that Step 6 in the following algorithm will be discussed in the next chapter, along with other algorithmic refinements.

Algorithm 3.2 (Trust Region Framework)

1. *Choose algorithm parameters*

   - *trust region acceptance values* b_1 *and* b_2 *satisfying* $0 \leq b_1 < b_2 \leq 1$,
   - *unboundness count* $b_{\max} > 0$,
   - *and the maximum trust region radius size* $\Delta_{\max} > 0$.

2. *For the starting point* w^0 *provided by Algorithm 3.1, select* $\Delta^0 \in (0, \Delta_{\max}]$.

3. *Set* $k = 0$.

4. *Identify activities at* w^k *and form* A^k*, the activity matrix consisting of the gradients of the activities at* w^k.

5. *Check for termination condition: If* w^k *is a second order point of* p_μ*, then terminate with* $w_\mu = w^k$.

6. *Find a "solution"* d_T^k *of*

$$TR(w^k, \Delta^k): \quad \min_d \quad \phi^k(d) \quad \textit{subject to} \quad \|d\|_\infty \leq \Delta^k.$$

7. *Evaluate a direction* d_T^k:

   (a) *Calculate* $\chi_a^k = p_\mu(w^k) - p_\mu(w^k + d_T^k)$.

   (b) *Calculate* $\chi_p^k = \phi^k(0) - \phi^k(d_T^k)$.

(c) *If* $\chi_a^k/\chi_p^k \geq b_2$*, then set* $w^{k+1} = w^k + d_T^k$ *and* $\Delta^{k+1} = \min(2\Delta^k, \Delta_{\max})$.

(d) *If* $b_1 \leq \chi_a^k/\chi_p^k < b_2$*, then set* $w^{k+1} = w^k + d_T^k$ *and* $\Delta^{k+1} = \Delta^k$.

(e) *Otherwise, set* $\Delta^k = \Delta^k/2$*. Repeat starting at Step 6.*

8. *If* $PF(\mu)$ *appears to be unbounded*

- *iterations* $k - b_{\max} + 1, \cdots, k$ *were very successful, and*
- *over iterations* $k - b_{\max} + 1, \cdots, k$, $(p_\mu(w^{i+1}) - F(w^{i+1}))/\mu$ *is not reduced,*

then terminate due to unboundedness, and set $w_\mu = w^{k-b_{\max}}$.

9. *Set* $k = k + 1$*. Repeat from Step 4.*

Chapter 4

Implementation Concerns

4.1 Introduction

To develop an effective implementation of Algorithm 3.2, the trust region algorithm requires some modifications from the stated description. These modifications, the focus of this chapter, affect three areas of the algorithms:

- Step 6: the calculation of the "solution" d_T of the trust region subproblem TR(w, Δ),
- Step 7: the evaluation of the direction d_T within the trust region framework, and
- Step 4: the identification of activities throughout the process.

These changes and the reasons they were required are described in this chapter.

4.2 Solving the Trust Region Subproblem

Each iteration of Algorithm 3.2 requires the solution of at least one instance of the trust region subproblem

$$\text{TR}(w, \Delta): \quad \min_{d} \quad \phi(d) \quad \text{subject to} \quad \|d\|_\infty \leq \infty,$$

for the current point w. Finding d_T, a local solution of TR, involves minimizing a piecewise quadratic function subject to simple bounds on the unknowns. This solution process is, by far, the most time consuming step of the algorithm for solving PF(μ).

Dennis and Schnabel (in Section 6.4 of [40]) have shown that it is not necessary to solve trust region subproblems exactly to get acceptable convergence results for trust region algorithms. In Chapter 5, we establish a similar result for our algorithm. The approximate solutions developed in this chapter satisfy the conditions required for convergence, as stated in the following chapter.

The choice of an approximate solution at the current point w depends on how w is classified relative to a minimum point of p_μ.

The following condition is assumed to be true throughout this work, and for the problems solved by the algorithm. Section 8.7 in [46] discusses issues relating to scaling.

Assumption 4.1 *The bilevel programming problem and its component functions are assumed to be well-scaled.*

4.2.1 Classifying the Current Point

The necessary optimality conditions for a minimum point of PF(μ), as expressed in Corollary 3.13, are used to classify the current point w. Before proceeding further, note that the following assumption applies to the entire chapter.

Assumption 4.2 *The penalty function subproblem is nondegenerate.*

Definition 4.1 *The values of γ, A, Z and H, as defined previously, are all calculated at the current point w.*

1. *A type one point is a point which is classified as being far from a stationary point of the penalty function.*

   Recall that stationary points satisfy $Z^T\gamma = 0$. Let $\Lambda > 0$ be the algorithm tolerance for determining closeness to a stationary point. The current point w is classified as a type one point if $\|Z^T\gamma\|_2 > \Lambda$.

2. *A type two point is a point which is classified as being close to a stationary point of the penalty function which is not also a first order point of p_μ.*

   Recall that first order points are stationary points which, along with multipliers Ψ, satisfy the conditions stated in Definition 3.3.4. The current point w is classified as a type two point if $\|Z^T\gamma\|_2 \leq \Lambda$, and the multiplier estimates, Ψ, calculated at w violate some of the conditions for a first order point of the penalty function. The vector Ψ, which approximates the multipliers at the nearby stationary point, is obtained as a least squares solution to the system of equations $A\Psi = \gamma$.

3. *A type three point is a point which is classified as being close to a first order point of the penalty function which is not a second order point of p_μ.*

   Recall that second order points are first order points at which the reduced Hessian matrix Z^THZ is positive semidefinite. The current point w is classified as a type three point if $\|Z^T\gamma\|_2 \leq \Lambda$ and the multiplier estimates defined at w satisfy all the conditions necessary at a first order point, but the matrix Z^THZ is not positive semidefinite.

4. *A type four point is a point which is classified as being close to a second order point of the penalty function.*

   The current point w is classified as a type four point if $\|Z^T\gamma\|_2 \leq \Lambda$, the multiplier estimates defined at w satisfy all the conditions necessary at a first order point, and the matrix Z^THZ is positive semidefinite.

Corollary 3.13 indicates that a type four point may be close to a minimum point of the penalty function. It is unlikely that any of the other types of points are close to a minimum point of p_μ, if properly classified.

The remainder of this section describes the desired properties of an approximate solution of the trust region subproblem at each of the four types of points. A technique is presented in each case to find a direction with these properties.

4.2.2 Approximating Solutions at Type One Points

Because a type one point w appears to be far from a stationary point, an approximate trust region solution at w, denoted d_1, should decrease the model function and satisfy the trust region constraint, while trying to move towards a stationary point. In order to achieve the latter goal, the current activities at w should be maintained at $w + d_1$, with the possibility that additional activities are picked up at $w + d_1$.

The activities at w are still active at $w + d_1$ (within the model), if $d_1 \in \mathcal{W}_1(w)$, where

$$
\begin{aligned}
\mathcal{W}_1(w) = \{ \quad & d \in \mathbb{R}^{n+m+p} : \\
& d^T\nabla G_i(w) + \frac{1}{2}d^T\nabla^2 G_i(w)d = 0, && i \in T^0, \\
& d^T\nabla\lambda_i(w) = 0, && i \in P^0_\lambda \cup P^0_=, \\
& d^T\nabla g_i(w) + \frac{1}{2}d^T\nabla^2 g_i(w)d = 0, && i \in P^0_g \cup P^0_=,
\end{aligned}
$$

$$\begin{aligned} &d^T(\nabla g_i(w) - \nabla\lambda_i(w)) + \frac{1}{2}d^T\nabla^2 g_i(w)d = 0, \quad i \in P'_=, \\ &d^T\nabla c_i(w) + \frac{1}{2}d^T\nabla^2 c_i(w)d = 0, \qquad\qquad i \in M^0\}. \end{aligned}$$

Considering only directions $d \in \mathcal{W}_1(w)$, the model function $\phi(d)$ can be rewritten. Defining $\varphi_1(d) = \phi(d)$ for $d \in \mathcal{W}_1(w)$, it follows that

$$\begin{aligned} \varphi_1(d) \;=\; & F(w) + d^T\nabla F(w) + \frac{1}{2}d^T\nabla^2 F(w)d \\ & -\mu \sum_{i\in T'} \min(G_i(w) + d^T\nabla G_i(w) + \frac{1}{2}d^T\nabla^2 G_i(w)d, 0) \\ & +\mu \sum_{i\in P'_\lambda\cup P'_g} |\min(\lambda_i(w) + d^T\nabla\lambda_i(w), g_i(w) + d^T\nabla g_i(w) + \frac{1}{2}d^T\nabla^2 g_i(w)d)| \\ & -\mu \sum_{i\in P^0_\lambda} \min(0, g_i(w) + d^T\nabla g_i(w) + \frac{1}{2}d^T\nabla^2 g_i(w)d) \\ & -\mu \sum_{i\in P^0_g} \min(\lambda_i(w) + d^T\nabla\lambda_i(w), 0) + \mu \sum_{i\in P'_=} |\lambda_i(w) + d^T\nabla\lambda_i(w)| \\ & +\mu \sum_{i\in M'} |c_i(w) + d^T\nabla c_i(w) + \frac{1}{2}d^T\nabla^2 c_i(w)d|. \end{aligned}$$

To maintain feasibility and provide descent, $d_1 \in \mathcal{W}_1$ should satisfy $\varphi_1(d_1) < 0$ and $\|d_1\|_\infty \leq \Delta$. In addition, d_1 should be relatively easy to calculate. Unfortunately, the quadratic constraints in $\mathcal{W}_1(w)$ are nontrivial to satisfy exactly and the set

$$\{d \in \mathcal{W}_1(w) : \|d\|_\infty \leq \Delta\}$$

may be empty. Therefore, the linear approximation $\mathcal{A}^T d_1 = 0$ of the constraint set is used instead, and the second order changes in the active penalty terms are ignored.

Any direction d that satisfies $\mathcal{A}^T d = 0$ can be written as $d = Zz$ where $\mathcal{A}^T Z = 0$, for some $z \in \mathbb{R}^{q-n_a}$, where $q = n+m+p$ and n_a is the number of active penalty terms at w (which is the number of columns in $\mathcal{A}$). Therefore, we restrict our attention to directions in this reduced space. We wish to find z_1 such that $\varphi_1(Zz_1) < 0$ and $\|Zz_1\|_\infty \leq \Delta$. Note that the trust region distance constraint is now in the form of general linear constraints which can always be satisfied.

The following result assists in the calculation of an approximate solution d_1.

Lemma 4.1 *For any $z \in \mathbb{R}^{q-n_a}$, there exists $\alpha_3 > 0$ such that, for $0 \leq \alpha_3$,*

$$\varphi_1(\alpha Z z) = \varphi_1(0) + \alpha z^T (Z^T \gamma) + \frac{1}{2}\alpha^2 z^T (Z^T B Z) z.$$

Proof: Choose α_3 small enough so that all the following conditions are satisfied for $0 \leq \alpha \leq \alpha_3$,

- for $i \in T'$,

$$\text{neg}[G_i(w) + \alpha z^T Z^T \nabla G_i(w) + \frac{1}{2}\alpha^2 z^T Z^T \nabla^2 G_i(w) Z z] = \text{neg}[G_i(w)],$$

- for $i \in P'_\lambda \cup P'_g$,

$$\text{sign}[g_i(w) - \lambda_i(w) + \alpha z^T Z^T (\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 z^T Z^T \nabla^2 g_i(w) Z z]$$
$$= \text{sign}[g_i(w) - \lambda_i(w)],$$

- for $i \in P'_\lambda \cup P'_=$,

$$\text{sign}[\lambda_i(w) + \alpha z^T Z^T \nabla \lambda_i(w)] = \text{sign}[\lambda_i(w)],$$

- for $i \in P'_g$,

$$\text{sign}[g_i(w) + \alpha z^T Z^T \nabla g_i(w) + \frac{1}{2}\alpha^2 z^T Z^T \nabla^2 g_i(w) Z z] = \text{sign}[g_i(w)],$$

- for $i \in P^0_\lambda$,

$$g_i(w) + \alpha z^T Z^T \nabla g_i(w) + \frac{1}{2}\alpha^2 z^T Z^T \nabla^2 g_i(w) Z z > 0,$$

- for $i \in P_g^0$,

$$\lambda_i(w) + \alpha z^T Z^T \nabla \lambda_i(w) > 0,$$

- for $i \in M'$,

$$\text{sign}[c_i(w) + \alpha z^T Z^T \nabla c_i(w) + \frac{1}{2}\alpha^2 z^T Z^T \nabla^2 c_i(w) Z z] = \text{sign}[c_i(w)].$$

The result follows immediately from the continuity of the inactive penalty terms over the interval, and from the definitions of γ and B at w. □

We will now consider $z_c = -Z^T\gamma$, the direction of steepest descent for φ_1 in the reduced space.

Definition 4.2 *The step $d_c = -\alpha_c Z Z^T \gamma$ from w is called the generalized Cauchy step of the trust region model. The step α_c is the first local minimum of $\varphi_1(\alpha Z z_c)$ within the trust region, that is the first local solution of the univariate minimization problem*

$$\min_{\alpha} \quad \varphi(\alpha Z z_c) \quad \textit{subject to} \quad 0 < \alpha \leq \alpha_\Delta,$$

where $\alpha_\Delta = \Delta/\|Zz_c\|_\infty$. The point $w + d_c$ is called the generalized Cauchy point.

The value of α_c can be calculated in a straightforward manner. The function $\varphi_1(\alpha Z z_c)$ is a piecewise quadratic function over the interval $\alpha \in (0, \alpha_\Delta]$, where $\alpha_\Delta = \Delta/\|Zz_c\|_\infty$. Let $\beta_1, \cdots \beta_t$ be the breakpoints, distinct values of α over $(0, \alpha_\Delta]$ at which the shape or differentiability of φ changes. Therefore, the β_i terms are positive values of α which satisfy one of the following equations:

- for $i \in T'$,

$$G_i(w) + \alpha z_c^T Z^T \nabla G_i(w) + \frac{1}{2}\alpha^2 z_c^T Z^T \nabla^2 G_i(w) Z z_c = 0,$$

- for $i \in P'_\lambda \cup P'_g \cup P^0_g \cup P'_=$,

$$\lambda_i(w) + \alpha z_c^T Z^T \nabla \lambda_i(w) = 0,$$

- for $i \in P'_\lambda \cup P'_g \cup P^0_\lambda \cup P'_=$,

$$g_i(w) + \alpha z_c^T Z^T \nabla g_i(w) + \frac{1}{2}\alpha^2 z_c^T Z^T \nabla^2 g_i(w) Z z_c = 0,$$

- for $i \in P'_\lambda \cup P'_g$,

$$(g_i(w) - \lambda_i(w)) + \alpha z_c^T Z^T (\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 z_c^T Z^T \nabla^2 g_i(w) Z z_c = 0,$$

- for $i \in M'$,

$$c_i(w) + \alpha z_c^T Z^T \nabla c_i(w) + \frac{1}{2}\alpha^2 z_c^T Z^T \nabla^2 c_i(w) Z z_c = 0.$$

Without loss of generality, assume that the breakpoints satisfy

$$0 = \beta_0 < \beta_1 < \cdots < \beta_t < \beta_{t+1} = \alpha_\Delta.$$

Due to the piecewise quadratic nature of φ_1, we can write

$$\varphi_1(\alpha Z z_c) = r_1^i + \alpha r_2^i + \frac{1}{2}\alpha^2 r_3^i = \theta_i(\alpha)$$

over the interval $\alpha \in (\beta_i, \beta_{i+1})$ for $i = 0 : t$. From Lemma 4.1, it follows that

$$r_1^0 = \varphi(0), \;\; r_2^0 = -\alpha \|Z^T \gamma\|_2^2 \text{ and } r_3^0 = z_c^T Z^T B Z z_c.$$

The values of r_j^{i+1}, for $j = 1, 2, 3$ and $i = 0 : t-1$, can be calculated from r_j^i by noting the changes in the penalty terms at β_{i+1}.

The generalized Cauchy step can be identified by analyzing φ_1 over each (β_i, β_{i+1}) interval. If

$$\left.\frac{\partial \varphi_1}{\partial \alpha}\right|_{\alpha=\beta_i} = r_2^i + \beta_i r_3^i \geq 0,$$

then z_c is not a descent direction over the interval $\alpha \in (\beta_i, \beta_{i+1})$. Therefore, no further analysis is necessary, since $\alpha_c = \beta_i$ corresponds to the first minimum along z_c. When z_c is a descent direction in the current interval, then further action is required. The minimum of the quadratic term $\theta_i(\alpha)$ occurs at $\alpha_{\min} = -r_3^i/r_2^i$. If $\alpha_{\min} \in (\beta_i, \beta_{i+1})$, then $\alpha_c = \alpha_{\min}$ corresponds to a minimum of φ along z_c. Otherwise, the next interval $(\beta_{i+1}, \beta_{i+2})$ is examined. If z_c is a descent direction over (β_t, α_Δ), but the minimum of $\theta_t(\alpha)$ does not lie in the interval, then we set $\alpha_c = \alpha_\Delta$. Note that, since r_2^0 is always negative, α_c always has a positive value.

While $d_1 = d_c$ is an acceptable trust region direction, it may be possible to find a direction that gives better descent. Let the matrix $\mathcal{A}_c$ be the activity matrix at w augmented with the gradients (already evaluated at w) of the activities picked up (within the model) at $w + d_c$. The values of γ_c and B_c are defined from γ and B using the derivatives evaluated at w for the new activities. Also, determine a matrix Z_c corresponding to $\mathcal{A}_c$. Motivated by the theory explained in Section 4.2.5, the trust region step will be improved, if the matrix $Z_c^T B_c Z_c$ is positive semidefinite, by using the quasi-Newton step $d_+ = Zz_+$, where z_+ is a least squares solution of

$$Z_c^T B_c Z_c z_+ = -Z_c^T \gamma_c.$$

If the direction $d_c + d_+$ satisfies

$$\varphi_1(d_c + d_+) < \varphi_1(d_c) \text{ and } \|d_c + d_+\|_\infty \le \Delta,$$

(that is, if d_+ further decreases the model function and $d_c + d_+$ lies within the current trust region), then accept $d_1 = d_c + d_+$ as the approximate trust region solution at the current type one point. Otherwise, $d_1 = d_c$ is viewed as the approximate trust region solution.

4.2.3 Approximating Solutions at Type Two Points

The multiplier estimates at w, a type two point, suggest that a nearby stationary point is not a first order point of the penalty function, and therefore, not a solution of PF(μ). An approximate trust region solution at w, denoted d_2, should decrease the model function and satisfy the trust region constraint, while moving away from this neighborhood. In order to achieve the latter goal, the activities whose multipliers are out of kilter (that is, the multipliers that violate the conditions for a first order point) are examined. A subset of these activities, denoted $\mathcal{D}$, will be dropped and the remaining activities will be maintained. Direction d_2 should be defined accordingly.

Let the set $\mathcal{D}_S$, for an activities index set S, denote the activities in S which are in $\mathcal{D}$. In particular, let $\mathcal{D}_{P^0_=}$ denote the indices $i \in P^0_=$ for which both $\lambda_i(w)$ and $g_i(w)$ are being dropped along d_2. Assume that both λ_i and g_i are being dropped only if both Ψ_i^λ and Ψ_i^g are in kilter, but their sum is not. Also, let $\mathcal{D}^\lambda_{P^0_=}$ denote the indices $i \in P^0_=$ for which $\lambda_i(w)$ is being dropped and $g_i(w)$ maintained along d_2. Similarly, define the set $\mathcal{D}^g_{P^0_=}$ for the opposite situation.

Ideally, $d_2 \in \mathcal{W}_2(w, \mathcal{D})$, where

$$
\begin{array}{llll}
\mathcal{W}_2(w,\mathcal{D}) = \{ & d \in I\!R^{n+m+p} : & & \\
& d^T\nabla G_i(w) + \frac{1}{2}d^T\nabla^2 G_i(w)d & = 0, & i \in S^0, \\
& d^T\nabla\lambda_i(w) & = 0, & i \in Q^0_\lambda \cup Q^0_=, \\
& d^T\nabla g_i(w) + \frac{1}{2}d^T\nabla^2 g_i(w)d & = 0, & i \in Q^0_g \cup Q^0_=, \\
& d^T(\nabla g_i(w) - \nabla\lambda_i(w)) + \frac{1}{2}d^T\nabla^2 g_i(w)d & = 0, & i \in Q'_=, \\
& d^T\nabla c_i(w) + \frac{1}{2}d^T\nabla^2 c_i(w)d & = 0, & i \in N^0 \quad \},
\end{array}
$$

and

$$S^0 = T^0 \setminus \mathcal{D}_{T^0}$$

$$
\begin{aligned}
Q^0_\lambda &= (P^0_\lambda \setminus \mathcal{D}_{P^0_\lambda}) \cup \mathcal{D}^g_{P^0_=} \\
Q^0_g &= (P^0_g \setminus \mathcal{D}_{P^0_g}) \cup \mathcal{D}^\lambda_{P^0_=} \\
Q'_= &= (P'_= \setminus \mathcal{D}_{P'_=}) \cup \mathcal{D}_{P^0_=} \\
Q^0_= &= P^0_= \setminus (\mathcal{D}_{P^0_=} \cup \mathcal{D}^\lambda_{P^0_=} \cup \mathcal{D}^g_{P^0_=}) \\
N^0 &= M^0 \setminus \mathcal{D}_{M^0}.
\end{aligned}
$$

The requirement for $i \in \mathcal{D}_{P^0_=}$ is not strictly necessary, but it reduces the number of directions which will be considered as approximate trust region solutions. It forces the activity $g_i - \lambda_i$ to be maintained (within the model) along the direction d_2 when both λ_i and g_i are being dropped.

The model function can be written over fewer terms by considering only $d \in \mathcal{W}_2(w, \mathcal{D})$. We denote $\phi(d)$ with d restricted to $\mathcal{W}_2(w, \mathcal{D})$ as $\varphi_2(d)$.

$$
\begin{aligned}
\varphi_2(d) = \; & F(w) + d^T \nabla F(w) + \frac{1}{2} d^T \nabla^2 F(w) d \\
& -\mu \sum_{i \in S'} \min(G_i(w) + d^T \nabla G_i(w) + \frac{1}{2} d^T \nabla^2 G_i(w) d, 0) \\
& +\mu \sum_{i \in Q_1} |\min(\lambda_i(w) + d^T \nabla \lambda_i(w), g_i(w) + d^T \nabla g_i(w) + \frac{1}{2} d^T \nabla^2 g_i(w) d)| \\
& -\mu \sum_{i \in Q^0_\lambda} \min(0, g_i(w) + d^T \nabla g_i(w) + \frac{1}{2} d^T \nabla^2 g_i(w) d) \\
& -\mu \sum_{i \in Q^0_g} \min(\lambda_i(w) + d^T \nabla \lambda_i(w), 0) + \mu \sum_{i \in Q'_=} |\lambda_i(w) + d^T \nabla \lambda_i(w)| \\
& +\mu \sum_{i \in N'} |c_i(w) + d^T \nabla c_i(w) + \frac{1}{2} d^T \nabla^2 c_i(w) d|.
\end{aligned}
$$

where

$$
\begin{aligned}
S' &= T' \cup \mathcal{D}_{T^0} \\
Q_1 &= P'_\lambda \cup P'_g \cup \mathcal{D}_{P'_=} \cup \mathcal{D}_{P^0_g} \cup \mathcal{D}_{P^0_\lambda} \\
N' &= M' \cup \mathcal{D}_{M^0}.
\end{aligned}
$$

Note that $\mathcal{W}_2(w, \mathcal{D})$ has the same form as $\mathcal{W}_1(w)$, defined for type one points, with S^0, Q^0_λ, Q^0_g, $Q'_=$, $Q^0_=$ and N^0 replacing T^0, P^0_λ, P^0_g, $P'_=$, $P^0_=$ and M^0, respectively. As well, $\varphi_2(d)$ for $d \in \mathcal{W}_2(w, \mathcal{D})$, and $\varphi_1(d)$ for $d \in \mathcal{W}_1(w)$, have similar forms, with S', Q_1 and N' replacing T', $P'_\lambda \cup P'_g$ and M', respectively.

As was the case for type one points, even if $\{d \in \mathcal{W}_2(w, \mathcal{D}) : \|d\|_\infty \leq \Delta\}$ is nonempty, satisfying $d \in \mathcal{W}_2(w, \mathcal{D})$ can be time consuming. Again, a first order approximation is used, and higher order change in the maintained activities is ignored.

Let $\mathcal{A}_\mathcal{D}$ be the activity matrix $\mathcal{A}$ with the columns corresponding to the gradients of the activities in $\mathcal{D}$ removed. The matrix $Z_\mathcal{D}$, satisfying $\mathcal{A}_\mathcal{D}^T Z_\mathcal{D} = 0$ and $Z_\mathcal{D}^T Z_\mathcal{D} = I$ can be defined from Z using $\mathcal{A}_\mathcal{D}$. It will have $n_\mathcal{D} = q - n_a + |\mathcal{D}|$, or $|\mathcal{D}|$ more columns than Z, where n_a is again the number of columns in $\mathcal{A}$. Linearizing the constraints defining $d \in \mathcal{W}_2(w, \mathcal{D})$ yields $\mathcal{A}_\mathcal{D}^T d = 0$, or equivalently, that there exists $z \in \mathbb{R}^{n_\mathcal{D}}$ such that $d = Z_\mathcal{D} z$. Thus, we are looking for z in the reduced space such that $\varphi_2(Z_\mathcal{D} z) < 0$ and $\|Z_\mathcal{D}^T z\|_\infty \leq \Delta$.

The approximate solution to the trust region subproblem is defined using the conditions stated in Lemma 3.9, which details how to define a descent direction for p_μ from a nondegenerate, stationary, non-first order point by dropping a single activity. This result can be generalized to dropping multiple activities from a type two point to find a descent direction for φ_2.

First, we state formally some assumptions which are placed on the choice of $\mathcal{D}$.

Assumption 4.3 *Both λ_i and g_i, for some $i \in P^0_=$, are in $\mathcal{D}$ only if Ψ^λ_i and Ψ^g_i are in kilter, but $\Psi^\lambda_i + \Psi^g_i$ is not. In addition, while both may be dropped, the difference $g_i - \lambda_i$ will be maintained along the dropping direction.*

Before verifying that multiple activities can be dropped from the current type two point, preliminary results and definitions are needed.

Lemma 4.2 *There exists $\alpha_4 > 0$ small enough such that all the following conditions are satisfied for $\alpha \in [0, \alpha_4]$ and any direction $d_{\mathcal{D}}$.*

- $i \in T'$:

$$neg[G_i(w) + \alpha d_{\mathcal{D}}^T \nabla G_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_i(w) d_{\mathcal{D}}] = neg[G_i(w)]$$

- $i \in \mathcal{D}_{T^0}$:

$$neg[\alpha d_{\mathcal{D}}^T \nabla G_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_i(w) d_{\mathcal{D}}] = neg[d_{\mathcal{D}}^T \nabla G_i(w)]$$

- $i \in P'_\lambda \cup P'_g$:

$$\begin{aligned} & neg[g_i(w) - \lambda_i(w) + \alpha d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}] \\ = \; & neg[g_i(w) - \lambda_i(w)] \end{aligned}$$

- $i \in P'_\lambda$:

$$sign[\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)] = sign[\lambda_i(w)]$$

- $i \in P'_g$:

$$sign[g_i(w) + \alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}] = sign[g_i(w)]$$

- $i \in P'_= \cup \mathcal{D}_{P'_=}$:

$$sign[\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)] = sign[\lambda_i(w)]$$

- $i \in \mathcal{D}_{P'_=}$:

$$\begin{aligned} & sign[\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w) + \\ & \quad \min(0, \alpha d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}})] = \; sign[\lambda_i(w)] \end{aligned}$$

  and

$$neg[\alpha d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}] = neg[d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w))]$$

- $i \in P_\lambda^0 \setminus \mathcal{D}_{P_\lambda^0}$:

$$g_i(w) + \alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}} > 0$$

- $i \in \mathcal{D}_{P_\lambda^0}$:

$$g_i(w) + \alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}} > \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)$$

  and

$$sign[\alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)] = sign[d_{\mathcal{D}}^T \nabla \lambda_i(w)]$$

- $i \in P_g^0 \setminus \mathcal{D}_{P_g^0}$:

$$\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w) > 0$$

- $i \in \mathcal{D}_{P_g^0}$:

$$\alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}} < \lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)$$

  and

$$sign[\alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}] = sign[d_{\mathcal{D}}^T \nabla g_i(w)]$$

- $i \in \mathcal{D}_{P_=^0}^\lambda$:

$$neg[\alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)] = neg[d_{\mathcal{D}}^T \nabla \lambda_i(w)]$$

- $i \in \mathcal{D}_{P_=^0}^g$:

$$neg[\alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}] = neg[d_{\mathcal{D}}^T \nabla g_i(w)]$$

- $i \in M'$:

$$sign[c_i(w) + \alpha d_{\mathcal{D}}^T \nabla c_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_i(w) d_{\mathcal{D}}] = sign[c_i(w)]$$

- $i \in \mathcal{D}_{M^0}$:

$$sign[\alpha d_{\mathcal{D}}^T \nabla c_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_i(w) d_{\mathcal{D}}] = sign[d_{\mathcal{D}}^T \nabla c_i(w)].$$

Proof: Follows immediately from the continuity of the functions. □

Definition 4.3 *Direction $d_{\mathcal{D}}$, defined under Assumption 4.3, is the direction*

$$d_{\mathcal{D}} = \sum_{i \in \mathcal{D}} \nu_i z_i$$

where z_i satisfies

$$z_i^T z_i = 1, \; z_i^T Z = 0 \text{ and } z_i^T \mathcal{A}_{\{i\}} = 0,$$

for $\mathcal{A}_{\{i\}}$, the matrix $\mathcal{A}$ with the gradient a_i corresponding to activity $i \in \mathcal{D}$ removed. The coefficients ν_i satisfy

$$\nu_i = \sigma_i / a_i^T z_i,$$

where $\sigma_i = \pm 1$ as indicated in the proof of Lemma 3.9 for each activity $i \in \mathcal{D}$ being dropped.

Notation 4.1 *For simplicity, the summation $\sum_{i \in S}$, for a set S, will occasionally be written as $\sum_S$ when it is clear that the summation is over index $i \in S$.*

Lemma 4.3 *The direction $d_{\mathcal{D}}$ and the gradient γ defined at w satisfy*

$$\begin{aligned} d_{\mathcal{D}}^T \gamma \;=\; & -\sum_{\mathcal{D}_{T^0}} |\Psi_i^G| - \sum_{\mathcal{D}^{\lambda}_{P^0_=}} |\Psi_i^{\lambda}| - \sum_{\mathcal{D}^{g}_{P^0_=}} |\Psi_i^{g}| - \sum_{\mathcal{D}_{P^-_=}} |\Psi_i^{g-\lambda}| - \sum_{\mathcal{D}_{P^{\pm}_=} \setminus \mathcal{D}^0_{P^{\pm}_=}} |\Psi_i^{g-\lambda}| - \sum_{\mathcal{D}^0_{P^{\pm}_=}} 0 \\ & - \sum_{\mathcal{D}_{P^0_g}} |\Psi_i^{g}| - \sum_{\mathcal{D}_{P^0_{\lambda}}} |\Psi_i^{\lambda}| - \sum_{\mathcal{D}_{P^0_=}} (\Psi_i^{\lambda} + \Psi_i^{g}) - \sum_{\mathcal{D}_{M^0}} |\Psi_i^{c}|, \end{aligned}$$

where

$$\mathcal{D}^0_{P^{\pm}_=} \;=\; \{ i \in \mathcal{D}_{P^{\pm}_=} : \Psi_i^{g-\lambda} = 0 \}.$$

Proof: Since Ψ is a least squares solution to the system $\mathcal{A}\Psi = \gamma$, there exists a vector ξ such that

$$\gamma = \mathcal{A}\Psi + Z\xi.$$

By the definition of $d_{\mathcal{D}}$, it follows that

$$
\begin{aligned}
d_{\mathcal{D}}^T \gamma &= \sum_{\mathcal{D}} \nu_i z_i^T (\mathcal{A}\Psi + Z\xi) \\
&= \sum_{\mathcal{D}} \nu_i z_i^T \mathcal{A}\Psi + \sum_{\mathcal{D}} \nu_i z_i^T Z\xi \\
&= \sum_{\mathcal{D}} \nu_i z_i^T \mathcal{A}\Psi + 0 \\
&= \sum_{\mathcal{D}} (\sigma_i / z_i^T a_i)(z_i^T a_i) e_i^T \Psi \\
&= \sum_{\mathcal{D}} \sigma_i \Psi_i \\
&= \sum_{\mathcal{D}_{T^0}} -\text{sign}[\Psi_i^G]\Psi_i^G + \sum_{\mathcal{D}^\lambda_{P^0_=}} -\text{sign}[\Psi_i^\lambda]\Psi_i^\lambda + \sum_{\mathcal{D}^g_{P^0_=}} -\text{sign}[\Psi_i^g]\Psi_i^g \\
&\quad + \sum_{\mathcal{D}_{P^-_=}} -\text{sign}[\Psi_i^{g-\lambda}]\Psi_i^{g-\lambda} + \sum_{\mathcal{D}_{P^\pm_=} \setminus \mathcal{D}^0_{P^\pm_=}} -\text{sign}[\Psi_i^{g-\lambda}]\Psi_i^{g-\lambda} + \sum_{\mathcal{D}^0_{P^\pm_=}} -1 \cdot 0 \\
&\quad + \sum_{\mathcal{D}_{P^0_g}} -\text{sign}[\Psi_i^g]\Psi_i^g + \sum_{\mathcal{D}_{P^0_\lambda}} -\text{sign}[\Psi_i^\lambda]\Psi_i^\lambda + \sum_{\mathcal{D}_{P^0_=}} (-\Psi_i^\lambda - \Psi_i^g) \\
&\quad + \sum_{\mathcal{D}_{M^0}} -\text{sign}[\Psi_i^c]\Psi_i^c \\
&= -\sum_{\mathcal{D}_{T^0}} |\Psi_i^G| - \sum_{\mathcal{D}^\lambda_{P^0_=}} |\Psi_i^\lambda| - \sum_{\mathcal{D}^g_{P^0_=}} |\Psi_i^g| - \sum_{\mathcal{D}_{P^-_=}} |\Psi_i^{g-\lambda}| - \sum_{\mathcal{D}_{P^\pm_=} \setminus \mathcal{D}^0_{P^\pm_=}} |\Psi_i^{g-\lambda}| - \sum_{\mathcal{D}^0_{P^\pm_=}} 0 \\
&\quad - \sum_{\mathcal{D}_{P^0_g}} |\Psi_i^g| - \sum_{\mathcal{D}_{P^0_\lambda}} |\Psi_i^\lambda| - \sum_{\mathcal{D}_{P^0_=}} (\Psi_i^\lambda + \Psi_i^g) - \sum_{\mathcal{D}_{M^0}} |\Psi_i^c|. \qquad \square
\end{aligned}
$$

Lemma 4.4 *Direction $d_{\mathcal{D}}$, as defined in Definition 4.3, provides descent for φ_2.*

Proof: Using Lemma 4.2, we have, for $0 \le \alpha \le \alpha_4$,

$$
\begin{aligned}
\varphi_2(\alpha d_{\mathcal{D}}) &= F(w) + \alpha d_{\mathcal{D}}^T \nabla F(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 F(w) d_{\mathcal{D}} \\
&\quad -\mu \sum_{T'} \text{neg}[G_i(w)](G_i(w) + \alpha d_{\mathcal{D}}^T \nabla G_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_i(w) d_{\mathcal{D}}) \\
&\quad -\mu \sum_{\mathcal{D}_{T^0}} \text{neg}[d_{\mathcal{D}}^T \nabla G_i(w)](\alpha d_{\mathcal{D}}^T \nabla G_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_i(w) d_{\mathcal{D}}) \\
&\quad +\mu \sum_{P'_\lambda} \text{sign}[\lambda_i(w)](\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w))
\end{aligned}
$$

$$
\begin{aligned}
&+\mu \sum_{P'_g} \text{sign}[g_i(w)](g_i(w) + \alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}) \\
&+\mu \sum_{\mathcal{D}_{P'_{=}}} \text{sign}[\lambda_i(w)]\lambda_i(w) \\
&+\mu \sum_{\mathcal{D}_{P'_{=}}} \text{sign}[\lambda_i(w)]\text{neg}[d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w))] \\
&\qquad (\alpha d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w)) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}) \\
&+\mu \sum_{\mathcal{D}_{P^0_\lambda}} \text{sign}[d_{\mathcal{D}}^T \nabla \lambda_i(w)]\alpha d_{\mathcal{D}}^T \nabla \lambda_i(w) \\
&+\mu \sum_{\mathcal{D}_{P^0_g}} \text{sign}[d_{\mathcal{D}}^T \nabla g_i(w)](\alpha d_{\mathcal{D}}^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}) \\
&-\mu \sum_{\mathcal{D}^\lambda_{P^0_{=}}} \text{neg}[d_{\mathcal{D}}^T \nabla \lambda_i(w)]\alpha d^T \nabla \lambda_i(w) \\
&-\mu \sum_{\mathcal{D}^g_{P^0_{=}}} \text{neg}[d_{\mathcal{D}}^T \nabla g_i(w)](\alpha d^T \nabla g_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_i(w) d_{\mathcal{D}}) \\
&+\mu \sum_{P'_{=} \setminus \mathcal{D}_{P'_{=}}} \text{sign}[\lambda_i(w)](\lambda_i(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_i(w)) \\
&+\mu \sum_{\mathcal{D}_{P^0_{=}}} \text{sign}[d_{\mathcal{D}}^T \nabla \lambda_i(w)]\alpha d_{\mathcal{D}}^T \nabla \lambda_i(w) \\
&+\mu \sum_{M'} \text{sign}[c_i(w)](c_i(w) + \alpha d_{\mathcal{D}}^T \nabla c_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_i(w) d_{\mathcal{D}}) \\
&+\mu \sum_{\mathcal{D}_{M^0}} \text{sign}[d_{\mathcal{D}}^T \nabla c_i(w)](\alpha d_{\mathcal{D}}^T \nabla c_i(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_i(w) d_{\mathcal{D}}) \\
= \;& \varphi_2(0) + \alpha d_{\mathcal{D}}^T \gamma + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}} d_{\mathcal{D}} \\
&-\mu\alpha \sum_{\mathcal{D}_{T^0}} \min(0, d_{\mathcal{D}}^T \nabla G_i(w)) - \mu\alpha \sum_{\mathcal{D}^\lambda_{P^0_{=}}} \min(d_{\mathcal{D}}^T \lambda_i(w), 0) \\
&-\mu\alpha \sum_{\mathcal{D}^g_{P^0_{=}}} \min(0, d_{\mathcal{D}}^T \nabla g_i(w)) \\
&+\mu\alpha \sum_{\mathcal{D}_{P'_{=}}} \text{sign}[\lambda_i(w)] \min(0, d_{\mathcal{D}}^T(\nabla g_i(w) - \nabla \lambda_i(w)))
\end{aligned}
$$

$$+\mu\alpha \sum_{\mathcal{D}_{P_g^0}} |d_{\mathcal{D}}^T \nabla g_i(w)| + \mu\alpha \sum_{\mathcal{D}_{P_\lambda^0}} |d_{\mathcal{D}}^T \nabla \lambda_i(w)|$$
$$+\mu\alpha \sum_{\mathcal{D}_{P_=^0}} |d_{\mathcal{D}}^T \nabla \lambda_i(w)| + \mu\alpha \sum_{\mathcal{D}_{M^0}} |d_{\mathcal{D}}^T \nabla c_i(w)|,$$

where

$$\begin{aligned} B_{\mathcal{D}} &= B - \mu \sum_{\mathcal{D}_{T^0}} \text{neg}[d_{\mathcal{D}}^T \nabla G_i(w)] \nabla^2 G_i(w) - \mu \sum_{\mathcal{D}^g_{P_=^0}} \text{neg}[d_{\mathcal{D}}^T \nabla g_i(w)] \nabla^2 g_i(w) \\ &+ \mu \sum_{\mathcal{D}_{P'_=}} \text{sign}[\lambda_i(w)] \text{neg}[d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w))] \nabla^2 g_i(w) \\ &+ \mu \sum_{\mathcal{D}_{P_g^0}} \text{sign}[d_{\mathcal{D}}^T \nabla g_i(w)] \nabla^2 g_i(w) + \mu \sum_{\mathcal{D}_{M^0}} \text{sign}[d_{\mathcal{D}}^T \nabla c_i(w)] \nabla^2 c_i(w). \end{aligned}$$

Combining the above expression with Lemma 4.3, we have that

$$\begin{aligned} \varphi_2(\alpha d_{\mathcal{D}}) &= \varphi_2(0) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}} d_{\mathcal{D}} \\ &- \alpha \sum_{\mathcal{D}_{T^0}} (|\Psi_i^G| + \mu \min(0, d_{\mathcal{D}}^T \nabla G_i(w))) \\ &- \alpha \sum_{\mathcal{D}^\lambda_{P_=^0}} (|\Psi_i^\lambda| + \mu \min(0, d_{\mathcal{D}}^T \nabla \lambda_i(w))) \\ &- \alpha \sum_{\mathcal{D}^g_{P_=^0}} (|\Psi_i^g| + \mu \min(0, d_{\mathcal{D}}^T \nabla g_i(w))) \\ &- \alpha \sum_{\mathcal{D}_{P_=^-}} (|\Psi_i^g| + \mu \min(0, d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w)))) \\ &- \alpha \sum_{\mathcal{D}_{P_=^+} \setminus \mathcal{D}^0_{P_=^+}} (|\Psi_i^g| - \mu \min(0, d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w)))) \\ &- \alpha \sum_{\mathcal{D}_{P_g^0}} (|\Psi_i^g| - \mu |d_{\mathcal{D}}^T \nabla g_i(w)|) - \alpha \sum_{\mathcal{D}_{P_\lambda^0}} (|\Psi_i^\lambda| - \mu |d_{\mathcal{D}}^T \nabla \lambda_i(w)|) \\ &- \alpha \sum_{\mathcal{D}_{P_=^0}} (\Psi_i^\lambda + \Psi_i^g - \mu |d_2^T \nabla \lambda_i(w)|) - \alpha \sum_{\mathcal{D}_{M^0}} (|\Psi_i^c| - \mu |d_{\mathcal{D}}^T \nabla c_i(w)|). \end{aligned}$$

Consider, for $i \in D_{T^0}$,

$$\begin{aligned} d_{\mathcal{D}}^T \nabla G_i(w) &= \sum_{j \in \mathcal{D}} \nu_j z_j^T \nabla G_i(w) = \nu_i z_i^T \nabla G_i(w) \\ &= (\sigma_i / z_i^T \nabla G_i(w)) z_i^T \nabla G_i(w) = \sigma_i = -\text{sign}[\Psi_i^G]. \end{aligned}$$

Similarly, for

$$
\begin{array}{lll}
i \in \mathcal{D}^{\lambda}_{P^0_=} : & d_{\mathcal{D}}^T \nabla \lambda_i(w) & = -\text{sign}[\Psi_i^{\lambda}] \\
i \in \mathcal{D}^{g}_{P^0_=} : & d_{\mathcal{D}}^T \nabla g_i(w) & = -\text{sign}[\Psi_i^{g}] \\
i \in \mathcal{D}_{P^-_=} : & d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w)) & = -\text{sign}[\Psi_i^{g-\lambda}] \\
i \in \mathcal{D}_{P^+_=} \setminus \mathcal{D}^0_{P^+_=} : & d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w)) & = -\text{sign}[\Psi_i^{g-\lambda}] \\
i \in \mathcal{D}^0_{P^+_=} : & d_{\mathcal{D}}^T (\nabla g_i(w) - \nabla \lambda_i(w)) & = -1 \\
i \in \mathcal{D}_{P^0_g} : & d_{\mathcal{D}}^T \nabla g_i(w) & = -\text{sign}[\Psi_i^{g}] \\
i \in \mathcal{D}_{P^0_\lambda} : & d_{\mathcal{D}}^T \nabla \lambda_i(w) & = -\text{sign}[\Psi_i^{\lambda}] \\
i \in \mathcal{D}_{P^0_=} : & d_{\mathcal{D}}^T \nabla \lambda_i(w) & = -\text{sign}[\Psi_i^{\lambda}] = -1 \\
i \in \mathcal{D}_{M^0} : & d_{\mathcal{D}}^T \nabla c_i(w) & = -\text{sign}[\Psi_i^{c}].
\end{array}
$$

Therefore, for $\alpha \in [0, \alpha_4]$:

$$\varphi_2(\alpha d_{\mathcal{D}}) = \varphi_2(0) - \alpha\vartheta + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}} d_{\mathcal{D}},$$

where

$$
\begin{aligned}
\vartheta = & \sum_{\mathcal{D}_{T^0}} (|\Psi_i^G| - \mu\text{pos}[\Psi_i^G]) + \sum_{\mathcal{D}^{\lambda}_{P^0_=}} (|\Psi_i^{\lambda}| - \mu\text{pos}[\Psi_i^{\lambda}]) + \sum_{\mathcal{D}^{g}_{P^0_=}} (|\Psi_i^{g}| - \mu\text{pos}[\Psi_i^{g}]) \\
& + \sum_{\mathcal{D}_{P^+_=}} (|\Psi_i^{g-\lambda}| - \mu\text{pos}[\Psi_i^{g-\lambda}]) + \sum_{\mathcal{D}_{P^-_=} \setminus \mathcal{D}^0_{P^+_=}} (|\Psi_i^{g-\lambda}| + \mu\text{pos}[\Psi_i^{g-\lambda}]) + \sum_{\mathcal{D}^0_{P^+_=}} \mu \\
& + \sum_{\mathcal{D}_{P^0_g}} (|\Psi_i^{g}| - \mu) + \sum_{\mathcal{D}_{P^0_\lambda}} (|\Psi_i^{\lambda}| - \mu) + \sum_{\mathcal{D}_{P^0_=}} (\Psi_i^{\lambda} + \Psi_i^{g} - \mu) + \sum_{\mathcal{D}_{M^0}} (|\Psi_i^{c}| - \mu) \\
= & \sum_{i \in \mathcal{D}} \vartheta_i.
\end{aligned}
$$

By examining each ϑ_i separately, and recalling that each multiplier being considered is out of kilter, it is easily shown that $\vartheta > 0$. There exists $\alpha_5 \in (0, \alpha_4)$ such that $\varphi_2(\alpha d_{\mathcal{D}}) < \varphi_2(0)$ for $\alpha \in (0, \alpha_5)$. Therefore, the direction $d_{\mathcal{D}}$ is a descent direction for the model function φ_2 at type two points. □

The approximate trust region solution d_2 is therefore defined to be $d_2 = \alpha_{\mathcal{D}} d_{\mathcal{D}}$, where $\alpha_{\mathcal{D}} > 0$ is the first local minimum of the univariate problem

$$\min_{\alpha} \quad \varphi(\alpha d_{\mathcal{D}}) \quad \text{subject to} \quad 0 < \alpha \leq \Delta / \|d_{\mathcal{D}}\|_\infty.$$

The value of $\alpha_{\mathcal{D}}$ can be determined by the same method used to calculate α_c for type one points.

4.2.4 Approximating Solutions at Type Three Points

By definition, a type three point appears to be close to a first order point which is not a second order point. The multiplier estimates Ψ indicate that the current activities may be active at a solution of $\mathrm{PF}(\mu)$. Recall that $\mathcal{W}_1$ was defined for type one points as the set of directions along which the current activities are maintained. Therefore, if we require that $d_3 \in \mathcal{W}_1$ for the approximate trust region direction d_3, the activities are maintained and the trust region objective function reduces to $\varphi_3(d) = \varphi_1(d)$.

As with the previous classes of points, considering only $d \in \mathcal{W}_1(w)$ requires satisfying quadratic constraints exactly, which, if possible, can be very costly. Once again, we simply require that the linear approximation $\mathcal{A}^T d = 0$ be satisfied while reducing $\varphi_3(d)$. Working in the reduced space associated with matrix Z, we wish to find a direction z which approximately solves

$$\min_{z} \quad \varphi_3(Zz) \quad \text{subject to} \quad \|Zz\|_\infty \leq \Delta.$$

From Lemma 4.1, we can write, for any direction z in the reduced space, and for small steps $\alpha > 0$,

$$\varphi_3(\alpha Z z) = \varphi_3(0) + \alpha z^T Z^T \gamma + \frac{1}{2}\alpha^2 z^T Z^T B Z z.$$

For type three points, the value of $Z^T\gamma$ satisfies $\|Z^T\gamma\| \leq \Lambda$ at w. At a type one point, the Cauchy direction $z = -Z^T\gamma$ provides reasonable descent for the model function. However, this direction could be of negligible size at a type three point. Therefore, second order information about w must be used in defining the approximate trust region solution d_3.

Motivated by the results of Corollary 3.13 and Lemmas 3.14, 3.16 and 3.17 for the penalty function, a direction z_{nc} of negative curvature of the reduced Hessian of the Lagrangian $Z^T HZ$ will be used to define d_3. The direction $d = Zz_{nc}$ may provide second order descent in φ_3, but may not provide first order descent. Define $d_{nc} = \sigma_{nc} Z z_{nc}$ where $\sigma_{nc} = \pm 1$ is defined so that $\sigma_{nc} z_{nc}^T Z^T \gamma < 0$ if $Z^T \gamma$ is nonzero. If $Z^T \gamma = 0$, then either value of σ_{nc} is acceptable.

Define $d_3 = \alpha_{nc} \sigma_{nc} Z z_{nc}$ where $\alpha_{nc} > 0$ is calculated as the minimum of the univariate problem

$$\min_{\alpha} \quad \varphi(\alpha \sigma_{nc} Z z_{nc}) \quad \text{subject to} \quad 0 < \alpha \leq \Delta / \|Z z_{nc}\|_\infty .$$

As explained on page 62 in Section 3.5.3, it is possible that $Z^T HZ$ is indefinite and d_{nc} does not provide descent. In this event, the algorithm currently terminates at such a point. In theory, using the result presented in Lemma 3.14, it is possible to find a descent direction if one exists. Such a technique was not investigated further.

4.2.5 Approximating Solutions at Type Four Points

As with type three points, the multiplier estimates at a type four point w indicate that a first order point appears to be nearby. However, unlike at a type three point, the curvature of the reduced Hessian of the Lagrangian at w, $Z^T HZ$, is positive semidefinite, which indicates that the nearby second order point may be a minimum point of the penalty function.

If the current point is really close to a second order point of p_μ, then a full Newton step should be taken. When started close to a solution, Newton's method will converge quickly to that solution.

A Newton step is composed of two orthogonal parts, the horizontal and vertical steps. The horizontal step maintains the current activities, up to first order change, while stepping to the true minimum of a quadratic approximation of the differentiable part of the penalty function. This step, d_h, satisfies $d_h = -Zz_h$, where z_h is the solution of system

$$Z^T HZz = -Z^T\gamma.$$

The step can be written as

$$d_h = -Z(Z^T HZ)^{-1}Z^T\gamma.$$

Because of the nonlinearity of the functions comprising the penalty function, it is unlikely that the activities at w are still active at $w+d_h$. We shall define a step d_v, called the vertical step, which is designed so that the activities at w (which the multipliers indicate may still be active at the nearby stationary point) are once again active at $w + d_h + d_v$. Let $\xi(w)$ be a vector of the active penalty terms at w, ordered as in $\mathcal{A}$. Therefore, $\xi(w+d_h)$ and $\xi(w+d_h+d_v)$ are these terms evaluated at $w+d_h$ and $w+d_h+d_v$, respectively. The Newton step solution to the system of equations $\xi(w + d_h + d_v) = 0$ is the step d satisfying

$$\xi(w + d_h) + d^T\nabla\xi(w + d_h) = 0.$$

Because of the cost of evaluating all the gradients at $w + d_h$, the matrix is approximated by its value at w, $\mathcal{A}(w)$. Therefore, we define the vertical step d_v as the solution of the system

$$\mathcal{A}(w)^T d_v = -\xi(w + d_h).$$

This direction can be calculated as

$$d_v = -\mathcal{A}(w)(\mathcal{A}(w)^T\mathcal{A}(w))^{-1}\xi(w + d_h).$$

Therefore, at point w, the steps d_h, d_v and d_N are calculated. The step d_N is considered a successful trust region step if

1. it falls within the current trust region, and

2. it decreases the penalty function from w.

Note that the second conditions requires a decrease in the penalty function, rather than a decrease in the model function along with a strong correspondence between the model and penalty functions. If w is truly close to a solution, the quadratic model is a very good approximation of the penalty function, so an explicit comparison is not necessary. Another way of viewing this requirement is that, near a solution, the model function used is the actual penalty function.

If the Newton step is unsuccessful (either because the step is too long or because it provides no decrease in the penalty function), then the algorithm concludes that w is not as close to a solution as originally thought, and the point was misclassified. In this case, the value of the closeness tolerance Λ is reduced, and the point is reclassified. If the step failed because the trust region radius was too small, the reduction of the tolerance will still allow the algorithm to approach a solution.

4.2.6 Comments on the Approximate Solutions

The approximate trust region solutions, as described in this section, are chosen so that they reduce the appropriate model function without too many expensive computations. Hopefully, at the same time, they will decrease the penalty function.

The goals for the approximate solutions are basically as follows.

- When far from a stationary point, the approximate solution is simply the generalized Cauchy point. The direction is defined using first order information, but the step size calculation involves first and second order information at the current point.

Using this simple direction, progress is likely to be made towards a stationary point.

- Once the current point appears to be close to a stationary point, the multiplier estimates become important. They indicate whether the current active set matches the active set at some second order point of the penalty function. If the current set of activities is not such a set, then the approximate trust region solution is determined via a dropping direction that is chosen to move away from the subspace defined by the current activities. Again, first order information is used to define the dropping direction, and first and second order information is used in the calculation of the step size. Progress towards a different stationary point is likely to be observed.

- When the current point appears to be close to a first order point, the approximate solution is chosen so that it stays within the subspace defined by the activities. By doing so, the linearization of the activities are maintained along the approximate solution. However, at type three points, while the active set seems to be correctly identified, the point w is not in a region of positive curvature. Therefore, the approximate trust region solution is based on a direction of negative curvature for the reduced Hessian of the Lagrangian at w. While maintaining the current activities (to first order change), progress is likely to be made towards a region in which the reduced Hessian is positive semidefinite.

- When the current point appears to be close to a second order point, a full Newton step is attempted. This direction attempts to move towards the minimum of the differentiable part of the penalty function and to maintain the current activities at the new point. If the Newton step fails, the algorithm concludes that the current point is not as close to a solution as thought, and w is reclassified.

4.3 Evaluating the Trust Region Direction

In the previous chapter, it was stated that a direction d_T is an acceptable trust region direction if there is reasonable agreement between the actual decrease observed along d_T in the penalty function and the decrease predicted by the model. Stated mathematically, it is required that

$$\chi_a(d_T)/\chi_p(d_T) > \kappa_1$$

for an algorithm parameter $\kappa_1 \in (0,1)$, where

$$\chi_a(d_T) = p_\mu(w) - p_\mu(w + d_T) \text{ and } \chi_p(d_T) = \varphi(0) - \varphi(d_T)$$

for the model function φ at points of type one, two, or three. Recall that with an appropriate updating of the activity index sets, φ_2 for type two points is of the same form as φ_1, and φ_3. Therefore, φ is used to signify the model function for these types of points.

The method used to define d_T ensures that $\chi_p(d_T) > 0$ is always satisfied. It follows that an acceptable trust region direction always reduces p_μ since $\chi_a(d_T) > 0$ must also be satisfied by an acceptable direction. While any amount of decrease may seem reasonable, a certain level of decrease must be observed in practice to guarantee the convergence of the algorithm to even a stationary point.

Therefore, d_T must pass an additional test before it can be considered an acceptable direction. It must provide sufficient descent (as defined in [46], page 100) in the penalty function.

Definition 4.4 *Let $r_1 \in (0,1]$ be an algorithm parameter. A direction d provides sufficient descent in the penalty function if*

$$p_\mu(w + d) - p_\mu(w) \leq r_1 d^T \gamma.$$

Note that the value of r_1 determines how much "sufficient decrease" is required for the direction. If r_1 is very small, only a small amount of decrease is actually required. As r_1 approaches one, more decrease is required for sufficient descent.

If d_T does not satisfy sufficient descent in the penalty function, then it is reclassified as an unacceptable trust region direction. In this case, the trust region radius Δ is reduced and the algorithmic process continues from the current point.

4.4 Recognizing Activities

The activities at a given point w correspond to the penalty terms which are nondifferentiable at w. In practice, many problems can arise using this precise definition of activities. The following points should be considered.

- Because the penalty terms are generally nonlinear, it may take many iterations before any of the terms are identically zero.
- When the magnitude of a penalty term is very small but nonzero, the term is differentiable at the current point. However, due to the continuity of the functions defining the penalty terms, there is a small region about the current point which contains at least one point of nondifferentiability.
- The approximate trust region direction d_T was defined so that the activities would still be active at the new point. Unfortunately, due to the nonlinearity of the penalty terms, it is unlikely that this goal is achieved.

Consider the following situation adapted from Conn (in [36]). For simplicity, assume that the penalty function has the form

$$p_\mu(w) = F(w) - \mu \min(G_1(w), 0) - \mu \min(G_2(w), 0).$$

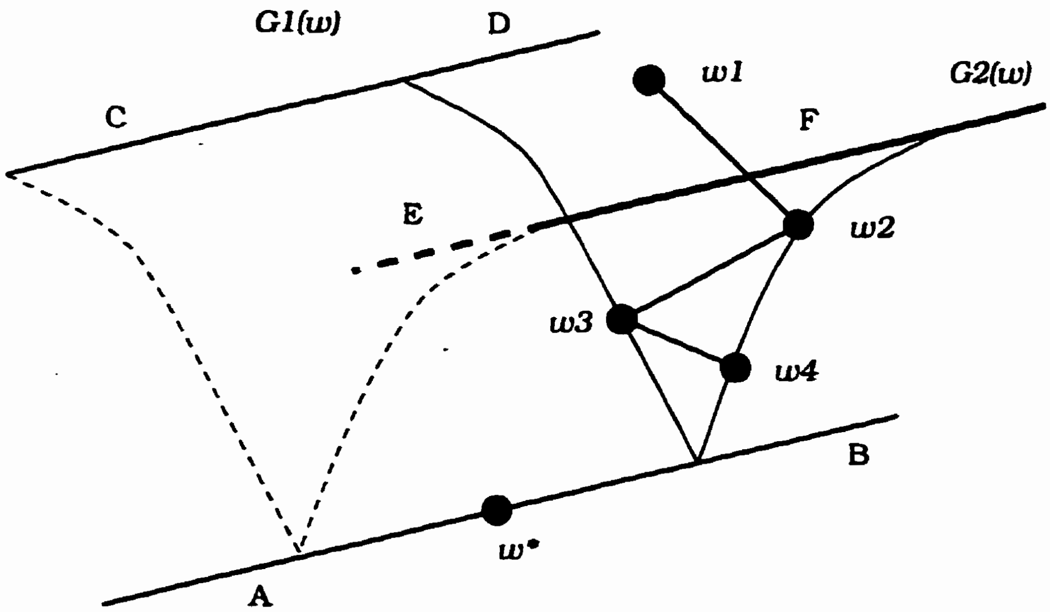

Figure 4.1: Motivation for ϵ-Activities

This objective function corresponds to a bilevel problem with two upper level constraints and an empty lower level problem (i.e. a standard nonlinear optimization problem).

The following comments make reference to Figure 4.1. Assume that w^*, at which $G_1(w^*) = G_2(w^*) = 0$, is a global solution of the penalty function. Note that the surface between AB and CD corresponds to the region in which $G_1(w) = 0$, and the surface between AB and EF is the region in which $G_2(w) = 0$. Consider minimizing p_μ, using the algorithm described thus far, starting from the point w_1, at which neither of the penalty terms is active. Minimization leads to the Cauchy point $w_2 = w_1 + \alpha_1^c d_1^c$, where

$$d_1^c = -\gamma_1 = -\left(\nabla F(w_1) - \mu\text{neg}[G_1(w_1)]\nabla G_1(w_1) - \mu\text{neg}[G_2(w_2)]\nabla G_2(w_1)\right).$$

At w_2, $G_2(w_2) = 0$ is satisfied, so $\mathcal{A} = [\nabla G_2(w_2)]$. Continuing the minimization from w_2 leads to $w_3 = w_2 + \alpha_2^c d_2^c$, where $d_2^c = -ZZ^T\gamma_2$ with $Z^T\mathcal{A} = 0$ and

$$\gamma_2 = \nabla F(w) - \mu\text{neg}[G_1(w_2)]\nabla G_1(w_2).$$

Now, while $G_1(w_3) = 0$, G_2 is no longer active, even though d_2^c was chosen orthogonal to $\nabla G_2(w_2)$.

The minimization process will continue to zigzag between the two constraint surfaces, with the magnitudes of both $G_1(w)$ and $G_2(w)$ becoming smaller. However, we will not reach a point at which both functions are exactly satisfied at the same time. To reach w^*, we need to project parallel to AB but a descent direction is never chosen in this direction. Therefore, we will never actually reach the solution w^*, although it will be reached, theoretically, in the limit.

The described behavior, which occurs in part because of the problems discussed above, illustrates the need for recognizing near activities as well as exact or true activities. Therefore, we introduce the concept of ϵ-activities, for some $\epsilon \geq 0$.

Definition 4.5

1. *For two values α and β, we define the following equivalence:*

$$\alpha \stackrel{\epsilon}{=} \beta \Leftrightarrow |\alpha - \beta| \leq \epsilon.$$

2. *The constraint set P is divided into three sets, for any value of ϵ:*

$$\begin{aligned} P_{=}(\epsilon) &= \{i \in P : \lambda_i(w) \stackrel{\epsilon}{=} g_i(w)\} \\ P_{\lambda}(\epsilon) &= \{i \in P \setminus P_{=}(\epsilon) : \lambda_i(w) < g_i(w)\} \\ P_{g}(\epsilon) &= \{i \in P \setminus P_{=}(\epsilon) : g_i(w) < \lambda_i(w)\}. \end{aligned}$$

3. *The following index sets define the ϵ-active penalty terms.*

$$\begin{aligned} T^0(\epsilon) &= \{i \in T : G_i(w) \stackrel{\epsilon}{=} 0\} \\ P_{\lambda}^0(\epsilon) &= \{i \in P_{\lambda}(\epsilon) : \lambda_i(w) \stackrel{\epsilon}{=} 0\} \end{aligned}$$

$$\begin{aligned} P_g^0(\epsilon) &= \{i \in P_g(\epsilon) : g_i(w) \stackrel{\epsilon}{=} 0\} \\ P_=^0(\epsilon) &= \{i \in P_=(\epsilon) : \lambda_i(w) \stackrel{\epsilon}{=} 0 \ or \ g_i(w) \stackrel{\epsilon}{=} 0\} \\ P_=^\prime(\epsilon) &= P_=(\epsilon) \setminus P_=^0(\epsilon) \\ M^0(\epsilon) &= \{i \in M : c_i(w) \stackrel{\epsilon}{=} 0\}. \end{aligned}$$

4. The following index sets define the ϵ-inactive penalty terms.

$$\begin{aligned} T'(\epsilon) &= T \setminus T^0(\epsilon) \\ P_\lambda'(\epsilon) &= P_\lambda(\epsilon) \setminus P_\lambda^0(\epsilon) \\ P_g'(\epsilon) &= P_g(\epsilon) \setminus P_g^0(\epsilon) \\ M'(\epsilon) &= M \setminus M^0(\epsilon). \end{aligned}$$

The use of ϵ-activities, rather than exact activities, has several important benefits. It means that penalty terms will be treated as activities more quickly, ϵ-inactive penalty terms are clearly differentiable in a larger region about the current point than exact activities, and the current ϵ-activities are much more likely to be ϵ-active at a new point than exactly active. By addressing these concerns, the behavior demonstrated in Figure 4.1 will be avoided. After a finite number of initial steps from w_1, the values of both G_1 and G_2 will be small enough to be considered ϵ-active. At that time, the correct active space would be identified and w^* reached because of the horizontal and vertical steps.

The existing trust region algorithm does not require many significant changes due to the introduction of ϵ-activities. Most of the changes simply involve replacing the exact activity sets with the defined ϵ-activity sets. However, note that the addition of the activity tolerance ϵ involves another parameter in the process of classifying points into the four groupings. This issue is dealt with in the next section.

The vertical step, used in defining d_T at type four points, is of added importance when ϵ-activities are used. This step was previously described as a way of mitigating the higher order change in the exact activities along the horizontal component of the Newton direction. As can now be seen, the step is also needed to bring the ϵ-activities closer to exact activities. If a penalty term is truly active at a minimum point of the penalty function, then it must eventually be exactly active within the algorithm as well. Because this is only important when approaching a solution, we do not consider vertical steps at any other stage of the algorithm.

4.5 Reclassifying the Current Point

As mentioned above, the algorithm parameters Λ and ϵ influence the classification of the current point w as a type one, two, three or four point. The value of Λ is used directly in the classification process. The value of ϵ determines which activities are considered active, and hence determines the values of $\mathcal{A}$, Z and γ.

Let w^S be the nearest stationary point of the penalty function in relation to w.

If $\|Z^T\gamma\|_2 < \Lambda$, then the algorithm uses the information available at w to approximate the values of the Lagrange multipliers at w^S. These approximations may be very inaccurate because w is not close enough to w^S (if the value of Λ is too large). Alternatively, the estimates could be poor because the set of ϵ-activities at w does not correspond to the exact activities at w^S. If some of the exact activities at w^S are not ϵ-active at w because their values are too large, then subsequent iterations will likely reduce their values so they will be considered ϵ-active at w. A more dangerous situation is that terms are considered ϵ-active at w when they are not active at w^S because the value of ϵ is too large. The information at w may be so different from the information at w^S that the multiplier estimates reveal no useful information about the behavior of the activities at w^S.

If $||Z^T\gamma||_2 > \Lambda$ even though the information at w may be very useful in predicting the multiplier values at w^S, the algorithm will not attempt to calculate the multiplier estimates. The value of ϵ or Λ may be too small to correctly classify w, but the generalized Cauchy direction will still provide descent in the model function. Therefore, the iterate will eventually be classified as close to stationary.

Inaccurate information about w^S derived from w when Λ or ϵ is too large may cause some problems within the algorithm. It could mean that a dropping direction, a direction of negative curvature, or a Newton direction does not provide any descent in the model function. This behavior would be detected by examining the solution of the univariate step length optimization problem. If a step of zero length is the solution at a type two, three, or four point, as described in the previous sections, then the algorithm concludes that Λ or ϵ is too large, and reduces both values.

If either ϵ or Λ is too small, increasing its value may improve the speed of convergence, but it is unlikely to change the outcome. Therefore, the algorithm does not attempt to detect this situation.

It is important to note that Λ and ϵ are reduced and w reclassified only when there is an indication that the current point is not classified correctly. In all other situations in which the approximate trust region direction is unacceptable, the radius is decreased and the algorithm continues from the current point.

4.6 Restatement of the Algorithm

The changes described in the previous sections are encorporated into the trust region algorithm stated below. A greater level of detail is included than in the previous statement of the algorithm.

Algorithm 4.1 (Revised Trust Region Framework)

1. *Choose algorithm parameters:*

   - *trust region acceptance values* b_1 *and* b_2 *satisfying* $0 \le b_1 < b_2 \le 1$,
   - *unboundness count* $b_{\max} > 0$,
   - *the maximum trust region radius size* $\Delta_{\max} > 0$, *and*
   - *sufficient decrease constant* $r_1 \in (0, 1]$.

2. *Choose initial starting values for algorithm tolerances:*

   - *closeness tolerance* $\Lambda^0 > 0$, *and*
   - *activity tolerance* $\epsilon^0 > 0$.

3. *For starting point* w^0 *provided by Algorithm 3.1, select* $\Delta^0 \in (0, \Delta_{\max}]$.

4. *Set* $k = 0$.

5. *Identify the* ϵ*-activities at* w^k *and form* $\mathcal{A}_\epsilon^k$, *the* ϵ*-activity matrix consisting of the gradients of the* ϵ*-activities.*

6. *Let* ξ^k *be the vector of values of the* ϵ*-activities, ordered as in* $\mathcal{A}_\epsilon^k$.

7. *Calculate* Z^k *satisfying* $(Z^k)^T Z^k = I$ *and* $(\mathcal{A}_\epsilon^k)^T Z^k = 0$.

8. *Calculate* γ^k *and* B^k *at* w^k*:*

$$
\begin{aligned}
\gamma^k &= \nabla F - \mu \sum_{T'(\epsilon)} neg[G_i] \nabla G_i + \mu \sum_{P'_\lambda(\epsilon)} sign[\lambda_i] \nabla \lambda_i + \mu \sum_{P'_g(\epsilon)} sign[g_i] \nabla g_i \\
&\quad + \mu \sum_{P'_{=}(\epsilon)} sign[\lambda_i] \nabla \lambda_i + \mu \sum_{M'(\epsilon)} sign[c_i] \nabla c_i \\
B^k &= \nabla^2 F - \mu \sum_{T'(\epsilon)} neg[G_i] \nabla^2 G_i + \mu \sum_{P'_g(\epsilon)} sign[g_i] \nabla^2 g_i + \mu \sum_{M'(\epsilon)} sign[c_i] \nabla^2 c_i.
\end{aligned}
$$

9. *Classify w^k:*

- *If* $\|(Z^k)^T\gamma^k\|_2 > \Lambda^k$, *then* w^k *is a Type One Point,*
- *Else*
  - *Find a least squares solution to* $\mathcal{A}_\epsilon^k \Psi^k = \gamma^k$.
  - *If* $\exists j$ *such that* Ψ_j^k *is out of kilter, then* w^k *is a Type Two Point.*
  - *Else*
    * *Calculate* H^k *at* w^k*:*

$$\begin{aligned} H^k &= B^k - \sum_{T^0(\epsilon)} \Psi_i^G \nabla^2 G_i - \sum_{P_g^0(\epsilon)} \Psi_i^g \nabla^2 g_i - \sum_{P_=^\ell(\epsilon)} \Psi_i^{g-\lambda} \nabla^2 g_i \\ &\quad - \sum_{P_=^0(\epsilon)} \Psi_i^g \nabla^2 g_i - \sum_{M^0(\epsilon)} \Psi_i^c \nabla^2 c_i. \end{aligned}$$

    * *If* $(Z^k)^T H^k Z^k$ *is indefinite, then* w^k *is a Type Three Point,*
    * *Else* w^k *is a Type Four Point.*

10. *Check for termination conditions:*

If w^k *is a Type Four Point at which* $(Z^k)^T\gamma^k = 0$ *and* $\xi^k = 0$, *then terminate the algorithm:* w^k *satisfies the necessary optimality conditions for a minimum point of* p_μ. *Set* $w_\mu = w^k$.

11. *Calculate* d_T^k, *an approximate solution of the trust region subproblem*

$$TR(w^k, \Delta^k): \ \min_d \ \varphi^k(d) \quad \text{subject to} \quad \|d\|_\infty \le \Delta^k.$$

- *If* w^k *is a Type Two Point, then*
  - *Choose a nonempty set* $\mathcal{D}$ *of* ϵ*-activities to be dropped from among those whose multipliers are out of kilter.*

– *Update the constraint sets using the dropping set $\mathcal{D}$.*

$$\begin{aligned}
T'(\epsilon) &= T'(\epsilon) \cup \mathcal{D}_{T^\epsilon} \\
T^0(\epsilon) &= T^0(\epsilon) \setminus \mathcal{D}_{T^\epsilon} \\
P'_\lambda(\epsilon) &= P'_\lambda(\epsilon) \cup \mathcal{D}_{P^\epsilon_\lambda} \cup \{i \in \mathcal{D}^\lambda_{P^\epsilon_=} : \Psi^\lambda_i > 0\} \\
P'_g(\epsilon) &= P'_g(\epsilon) \cup \mathcal{D}_{P^0_g(\epsilon)} \cup \{i \in \mathcal{D}^g_{P^0_=(\epsilon)} : \Psi^g_i > 0\} \\
P^0_\lambda(\epsilon) &= (P^0_\lambda(\epsilon) \setminus \mathcal{D}_{P^0_\lambda(\epsilon)}) \cup \{i \in \mathcal{D}^g_{P^0_=(\epsilon)} : \Psi^g_i < 0\} \\
P^0_g(\epsilon) &= (P^0_g(\epsilon) \setminus \mathcal{D}_{P^0_g(\epsilon)}) \cup \{i \in \mathcal{D}^\lambda_{P^0_=(\epsilon)} : \Psi^\lambda_i < 0\} \\
P'_=(\epsilon) &= (P'_=(\epsilon) \setminus \mathcal{D}_{P'_=(\epsilon)}) \cup \mathcal{D}_{P^0_=(\epsilon)} \\
P^0_=(\epsilon) &= P^0_=(\epsilon) \setminus \left(\mathcal{D}_{P^0_=(\epsilon)} \cup \mathcal{D}^\lambda_{P^0_=(\epsilon)} \cup \mathcal{D}^g_{P^0_=(\epsilon)}\right) \\
M'(\epsilon) &= M'(\epsilon) \cup \mathcal{D}_{M^0(\epsilon)} \\
M^0(\epsilon) &= M^0(\epsilon) \setminus \mathcal{D}_{M^0(\epsilon)}.
\end{aligned}$$

• *For type one, type two, and type three points, define the model function at w^k, letting $P_1(\epsilon) = P'_\lambda(\epsilon) \cup P'_g(\epsilon)$,*

$$\begin{aligned}
\varphi^k(\alpha d) &= F + \alpha d^T \nabla F + \frac{1}{2}\alpha^2 d^T \nabla^2 F d \\
&\quad -\mu \sum_{T'(\epsilon)} \min(G_i + \alpha d^T \nabla G_i + \frac{1}{2}\alpha^2 d^T \nabla^2 G_i d, 0) \\
&\quad -\mu \sum_{T^0(\epsilon)} \min(G_i, 0) \\
&\quad +\mu \sum_{P_1(\epsilon)} |\min(\lambda_i + \alpha d^T \nabla \lambda_i, g_i + \alpha d^T \nabla g_i + \frac{1}{2}\alpha^2 d^T \nabla^2 g_i d)| \\
&\quad +\mu \sum_{P^0_\lambda(\epsilon)} |\min(\lambda_i, g_i + \alpha d^T \nabla g_i + \frac{1}{2}\alpha^2 d^T \nabla^2 g_i d)| \\
&\quad +\mu \sum_{P^0_g(\epsilon)} |\min(\lambda_i + \alpha d^T \nabla \lambda_i, g_i)| \\
&\quad +\mu \sum_{P'_=(\epsilon)} |\lambda_i + \alpha d^T \nabla \lambda_i + \min(g_i - \lambda_i, 0)| + \mu \sum_{P^0_=(\epsilon)} |\min(\lambda_i, g_i)|
\end{aligned}$$

$$+\mu \sum_{M'(\epsilon)} |c_i + \alpha d^T \nabla c_i + \frac{1}{2}\alpha^2 d^T \nabla^2 c_i d| + \mu \sum_{M^0(\epsilon)} |c_i|.$$

- *If w^k is a Type One Point, then*
  - $d_c^k = -Z^k(Z^k)^T\gamma^k$ *and* $\alpha_c^k > 0$ *is the first local minimum of*

    $$TR_\alpha(d_c^k): \min_\alpha \quad \varphi^k(\alpha d_c^k) \quad \text{subject to} \quad 0 < \alpha \le \Delta^k/\|d_c^k\|_\infty.$$

  - *Augment A_ϵ^k with the gradients (evaluated at w^k) of the penalty terms which are ϵ-active at $w^k + \alpha_c^k d_c^k$ but not at w^k. Denote the augmented matrix by A_c^k.*
  - *Define Z_c^k satisfying $(Z_c^k)^T Z_c^k = I$ and $(Z_c^k)^T A_c^k = 0$.*
  - *Define γ_c^k by removing from γ^k the components due to the new ϵ-activities.*
  - *Similarly define B_c^k from B^k.*
  - *If $(Z_c^k)^T B_c^k Z_c^k$ is positive semidefinite, then*
    - *Calculate z_+^k, a least squares solution to $(Z_c^k)^T B_c^k Z_c^k z_+^k = -(Z_c^k)^T \gamma_c^k$.*
    - *If $\alpha_c^k d_c^k + Z_c^k z_+^k$ lies within the trust region and further decreases φ^k, then $d_T^k = \alpha_c^k d_c^k + Z_c^k z_+^k$,*
    - *Else $d_T^k = \alpha_c^k d_c^k$.*
  - *Else $d_T^k = \alpha_c^k d_c^k$.*
- *If w^k is a Type Two Point, then*
  - $d_{\mathcal{D}}^k = \sum_{i\in\mathcal{D}}(1/a_i^T z_i)\sigma_i z_i$, *where $\sigma_i = \pm 1$ from Lemma 3.9 depends on the type of ϵ-activity being dropped and a_i is the ith column of A_ϵ^k. The vector z_i^k satisfies $(z_i^k)^T z_i^k = 1$, $(Z^k)^T z_i^k = 0$, and $(A_i^k)^T z_i^k = 0$, where A_i^k is matrix A_ϵ^k with a_i removed.*
  - *Calculate $\alpha_{\mathcal{D}}^k$, the first local minimum of $TR_\alpha(d_{\mathcal{D}}^k)$.*
  - *If $\alpha_{\mathcal{D}}^k > 0$, then $d_T^k = \alpha_{\mathcal{D}}^k d_{\mathcal{D}}^k$. Otherwise, go to Step 15.*

- *If w^k is a Type Three Point, then*
  - *Calculate z^k_{nc}, a direction of negative curvature of $(Z^k)^T H^k Z^k$.*
  - *Choose $\sigma^k_{nc} = \pm 1$ such that $\sigma^k_{nc}((Z^k)^T z^k_{nc})^T \gamma^k \leq 0$.*
  - *Calculate α^k_{nc}, the first local minimum of $TR_\alpha(\sigma^k_{nc} Z^k z^k_{nc})$.*
  - *If $\alpha^k_{nc} > 0$, then $d^k_T = \alpha^k_{nc}\sigma^k_{nc} Z^k z^k_{nc}$. Otherwise, go to Step 15.*
- *If w^k is a Type Four Point, then*
  - *Calculate $d^k_h = -(Z^k)((Z^k)^T H^k (Z^k))^{-1}(Z^k)^T \gamma^k$.*
  - *Calculate $d^k_v = -(\mathcal{A}^k)((\mathcal{A}^k)^T \mathcal{A}^k)^{-1}\xi^k(w^k + d^k_h)$.*
  - *Calculate $d_N = d_h + d_v$.*
  - *Evaluate d^k_h:*
    - *Calculate $s^k = r_1(d^k_N)^T \gamma^k$.*
    - *If $p_\mu(w^k + d^k_N) - p_\mu(w^k) \leq s^k$ and $\|d^k_N\|_\infty \leq \Delta^k$, then $w^{k+1} = w^k + d^k_N$. Go to 14.*
    - *Else Go to 15.*

12. *Evaluate the direction d^k_T:*

- *Calculate $\chi^k_a = p_\mu(w^k + d^k_T) - p_\mu(w^k)$.*
- *Calculate $\chi^k_p = \varphi^k(d^k_T) - \varphi^k(0)$.*
- *Calculate $s^k = r_1 \gamma^T d^k_T$.*
- *If $\chi^k_a / \chi^k_p \geq b_2$ and $\chi^k_a \leq s^k$, then update*
  - *$w^{k+1} = w^k + d^k_T$.*
  - *If $\|d^k_T\|_\infty \geq \frac{1}{2}\Delta^k$, then $\Delta^{k+1} = \min(2\Delta^k, \Delta_{\max})$,*
  - *Else $\Delta^{k+1} = \Delta_k$.*
- *If $b_1 \leq \chi^k_a / \chi^k_p < b_2$ and $\chi^k_a \leq s^k$, then update*

- $w^{k+1} = w^k + d_T^k$,
- $\Delta^{k+1} = \Delta^k$.

* *Otherwise,* $\Delta^k = \Delta^k/2$ *and repeat from Step 11.*

13. *If PF(μ) appears to be unbounded, i.e.*

    * *iterations* $k - b_{\max} + 1, \cdots, k$ *were very successful, and*
    * *over iterations* $k - b_{\max} + 1, \cdots, k$, $p_\mu(w^{i+1}) - F(w^{i+1})$ *is not reduced,*

    then terminate due to unboundedness, and set $w_\mu = w^{k-b_{\max}}$.

14. *Update* $\epsilon^{k+1} = \epsilon^k$, $\Lambda^{k+1} = \Lambda^k$ *and* $k = k+1$. *Repeat from Step 5.*

15. *Reduce* ϵ^k *and* Λ^k. *Repeat from Step 5.*

Convergence results for this algorithm are presented in the next chapter.

Chapter 5

Convergence of the Algorithm

5.1 Introduction

In this chapter, common convergence analysis assumptions and several additional assumptions specific to the bilevel penalty function are used to establish that Algorithm 4.1 converges to a minimum point of the penalty function.

For convenience, the following definitions are repeated from Chapter 4.

- A type one point is classified as being far from a stationary point.
- A type two point is classified as being close to a stationary, non-first order point.
- A type three point is classified as being close to a first, non-second order point.
- A type four point is classified as being close to a second order point.

The convergence of the algorithm is established through a series of intermediate results, modeled after the convergence proofs presented in [34], [37] and [38].

1. When started far from a stationary point, the iterates generated by the algorithm approach a stationary point after a finite number of iterations, and the current point is eventually classified as a type two, three, or four point.

2. If the sequence of iterates is approaching a stationary, non-first order point, the current point is classified as a type two point after a finite number of iterations. The algorithm eventually identifies a successful dropping direction.

3. If the sequence of iterates is approaching a first order point, the current point is classified as a type three or four point after a finite number of iterations.

4. If the sequence of iterates is approaching a second order point, then the current point is classified as a type four point after a finite number of iterations. The algorithm eventually accepts a full Newton step.

5. Eventually, all iterations take successful Newton steps and convergence to a second order point of the penalty function is assured.

5.2 Assumptions and Terminology

The following assumptions and terminology are needed throughout this chapter.

Assumption 5.1

1. *The functions $F(w)$, $G_i(w)$ for $i \in T$, $g_i(w)$ for $i \in P$, and $c_i(w)$ for $i \in M$, are twice continuously differentiable.*

2. *The set*

$$\mathcal{F}^0 = \{w \in \mathbb{R}^{n+m+p} : p_\mu(w) < p_\mu(w^0)\},$$

for any starting value $w^0 \in \mathbb{R}^{n+m+p}$, is compact and has a nonempty interior.

It is necessary to introduce a measurement of the curvature, or second order change, of the penalty and model functions used in the algorithm. The measure used for differentiable functions is extended to handle nondifferentiabilities.

Definition 5.1

1. *The Generalized Rayleigh Quotient provides a measure of second and higher order change in a function f at a point w along a step αs. It is given by*

$$\Omega(f, w, \alpha s) = \frac{2}{\|\alpha s\|_2^2}\left(f(w+\alpha s) - f(w) - \alpha s^T \nabla f(w)\right).$$

If f is twice continuously differentiable, then this expression can be rewritten as

$$\Omega(f, w, \alpha s) = \frac{s^T \nabla^2 f(z) s}{\|s\|_2^2},$$

for some $z \in N(w, \alpha s) = \{w' : \|w - w'\| \leq \|w - \alpha s\|\}$. If f is a quadratic function, then $\nabla^2 f$ is a constant.

2. *The measure of curvature of the penalty function p_μ at w along the step αd is defined using the measures of curvature of its individual functions,*

$$\begin{aligned}\Omega_p(w, \alpha d) &= \Omega(F, w, \alpha d) - \mu \sum_{i \in T} \rho_i(\alpha)\Omega(G_i, w, \alpha d) \\ &\quad + \mu \sum_{i \in P} \sigma_i(\alpha)\rho_i(\alpha)\Omega(g_i, w, \alpha d) + \mu \sum_{i \in M} \sigma_i(\alpha)\Omega(c_i, w, \alpha d),\end{aligned}$$

where

$$\begin{aligned} i \in T, \quad \rho_i(\alpha) &= neg[G_i(w+\alpha d)] \\ i \in P, \quad \sigma_i(\alpha) &= sign[\min(\lambda_i(w+\alpha d), g_i(w+\alpha d))] \\ \rho_i(\alpha) &= neg[g_i(w+\alpha d) - \lambda_i(w+\alpha d)] \\ i \in M, \quad \sigma_i(\alpha) &= sign[c_i(w+\alpha d)]. \end{aligned}$$

3. *The measure of curvature for the piecewise quadratic model function φ^k defined at w^k along a step αd^k is defined using second order change in the penalty terms,*

$$\begin{aligned}\Omega^k(w^k,\alpha d^k) &= (d^k)^T\nabla^2 F(w^k)d^k - \mu\sum_{T'(\epsilon)}\varrho_i(\alpha d^k)(d^k)^T\nabla^2 G_i(w^k)d^k \\ &\quad -\mu\sum_{P_\lambda^\epsilon}\varrho_i(\alpha d^k)(d^k)^T\nabla^2 g_i(w^k)d^k \\ &\quad +\mu\sum_{P'_\lambda(\epsilon)\cup P'_g(\epsilon)}\varsigma_i(\alpha d^k)\varrho_i(\alpha d^k)(d^k)^T\nabla^2 g_i(w^k)d^k \\ &\quad +\mu\sum_{M'(\epsilon)}\varsigma_i(\alpha d^k)(d^k)^T\nabla^2 c_i(w^k)d^k,\end{aligned}$$

where

- *for $i\in T'(\epsilon)$,*

$$\varrho_i(\alpha d^k) = neg[G_i(w^k)+\alpha(d^k)^T\nabla G_i(w^k)+\frac{1}{2}\alpha^2(d^k)^T\nabla^2 G_i(w^k)d^k],$$

- *for $i\in P_\lambda^\epsilon$,*

$$\varrho_i(\alpha d^k) = neg[g_i(w^k)+\alpha(d^k)^T\nabla g_i(w^k)+\frac{1}{2}\alpha^2(d^k)^T\nabla^2 g_i(w^k)d^k],$$

- *for $i\in P'_\lambda(\epsilon)\cup P'_g(\epsilon)$,*

$$\begin{aligned}\varsigma_i(\alpha d^k) &= sign[\min(\lambda_i(w^k)+\alpha(d^k)^T\nabla\lambda_i(w^k), \\ &\qquad g_i(w^k)+\alpha(d^k)^T\nabla g_i(w^k)+\frac{1}{2}\alpha^2(d^k)^T\nabla^2 g_i(w^k)d^k)] \\ \varrho_i(\alpha d^k) &= neg[g_i(w^k)-\lambda_i(w^k)+\alpha(d^k)^T(\nabla g_i(w^k)-\nabla\lambda_i(w^k)) \\ &\qquad +\frac{1}{2}\alpha^2\nabla^2 g_i(w^k)d^k],\end{aligned}$$

- *for $i\in M'(\epsilon)$*

$$\varsigma_i(\alpha d^k) = sign[c_i(w^k)+\alpha(d^k)^T\nabla c_i(w^k)+\frac{1}{2}\alpha^2(d^k)^T\nabla^2 c_i(w^k)d^k].$$

4. *Let β_j^k be the s_c^k distinct breakpoints of φ^k along d_c^k. As in Section 4.2.2, assume that*

$$0 = \beta_0^k < \beta_1^k < \cdots < \beta_{s_c^k}^k < \beta_{s_c^k+1}^k = \alpha_\Delta^k.$$

Define the second order change in φ^k along d_c^k at each breakpoint as

$$\Omega_j^k = \Omega^k(w^k, \beta_j^k d_c^k), \qquad \text{for } j = 1 : s_c^k + 1,$$

In particular, note that

$$\Omega_1^k = \frac{(\gamma^k)^T Z^k (Z^k)^T B^k Z^k (Z^k)^T \gamma^k}{\|(Z^k)^T \gamma^k\|_2^2}.$$

5. *Let $\hat{\beta}_j^k$ be the s^k distinct breakpoints of φ^k along the approximate trust region solution d^k, as for β_j^k above. Similarly define the second order change in φ^k along d^k at each breakpoint as*

$$\zeta_j^k = \Omega^k(w^k, \hat{\beta}_j^k d^k), \text{ for } j = 1 : s^k.$$

6. *Define the following curvature measurements.*

$$\begin{aligned} \Omega^k &= \max_{j=1:s_c^k+1} |\Omega_j^k| \\ \zeta^k &= \max_{j=1:s^k+1} |\zeta_j^k| \\ \tau^k &= 1 + \max_{j=1:k} (\Omega^j, \zeta^j). \end{aligned}$$

The following assumption is similar to Assumption A5.4 in [38].

Assumption 5.2 *The curvature measurements satisfy the following condition over all iterations k,*

$$\sum_{k=1}^{\infty} \frac{1}{\tau^k} = +\infty.$$

Thus, there exists a finite value $\tau^{\max} > 1$ such that $\tau^k \le \tau^{\max}$ for all iterations k, i.e. τ^k is bounded above (by $\tau^{\max}$) and below (by one).

5.3 Convergence to a Near Stationary Point

The first stage of the convergence proof establishes that the algorithm converges to a near stationary point, as defined below.

Definition 5.2 *A near stationary point is a type two, three, or type four point, as defined at the beginning of this chapter.*

The result is proven in three distinct parts, as described below.

1. A lower bound is established for the change in the penalty function from a type one point w^k to a new point $w^k + d^k$, where d^k is a successful trust region direction.
2. By examining the difference between the values of the model function and the penalty function at $w^k + d^k$, and by establishing a lower bound on the size of the trust region radius over a sequence of type one points, the decrease in the penalty function is proven to be bounded away from zero.
3. The algorithm is then proven to approach a near stationary point.

5.3.1 A Bound on the Penalty Function Decrease

This section is modeled after Section 3.1 ("Obtaining a sufficient decrease in the model") in [38]. To establish a lower bound on the decrease in the penalty function along the trust region direction d^k defined from a type one point w^k, it is first necessary to obtain a lower bound on the decrease in the model function φ^k along d^k. Because d^k is chosen so that it provides at least as much decrease in φ^k as the generalized Cauchy step $\alpha_c^k d_c^k$, the bound can be obtained by analyzing the Cauchy step.

For a direction $d_c^k = -(Z^k)^T Z^k \gamma^k$, the step α_c^k can be defined in one of three ways:

- $\alpha_c^k = \alpha_0^{\min} \in (0, \beta_1^k)$, where $\alpha_0^{\min}$ is the minimum point of the quadratic form $q_0(\alpha) = \varphi^k(\alpha d_c^k)$ for $\alpha \in (0, \beta_1^k)$;
- $\alpha_c^k = \beta_1^k = \alpha_\Delta^k$, when there are no breakpoints along d_c^k in the interval $(0, \alpha_\Delta^k)$;
- $\alpha_c^k \geq \beta_1^k$, when there is at least one breakpoint $\beta_i \in (0, \alpha_\Delta^k)$.

A lower bound is obtained for $\varphi^k(0) - \varphi(\alpha_c^k d_c^k)$ by examining each way that α_c^k can be defined.

Lemma 5.1 *Given Assumptions 5.1 and a type one point w^k, if $\alpha_c^k = \alpha_0^{\min} \in (0, \beta_1^k)$, then*

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \frac{\|(Z^k)^T \gamma^k\|_2^2}{2\tau^k}.$$

Proof: Recall from Lemma 4.1 and the definition of Ω_1^k that for $\alpha \in (0, \beta_1^k)$,

$$\begin{aligned} q_0(\alpha) = \varphi^k(\alpha d_c^k) &= \varphi^k(0) - \alpha\|(Z^k)^T\gamma^k\|_2^2 + \frac{1}{2}\alpha^2(\gamma^k)^T Z^k (Z^k)^T B^k Z^k (Z^k)^T \gamma^k \\ &= \varphi^k(0) - \alpha\|(Z^k)^T\gamma^k\|_2^2 + \frac{1}{2}\alpha^2\|(Z^k)^T\gamma^k\|_2^2 \Omega_1^k. \end{aligned}$$

The minimum of q_0 occurs at $\alpha_0^{\min} = 1/\Omega_1^k$. Therefore, since $\alpha_0^{\min} \in (0, \beta_1^k)$,

$$q_0(\alpha_c^k) = \varphi^k(\alpha_0^{\min} d_c^k) = \varphi^k(0) - \frac{1}{2}\frac{\|(Z^k)^T\gamma^k\|_2^2}{\Omega_1^k}.$$

Therefore,

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) = \frac{\|(Z^k)^T\gamma^k\|_2^2}{2\Omega_1^k} \geq \frac{\|(Z^k)^T\gamma^k\|_2^2}{2\tau^k},$$

since $|\Omega_1^k| \leq \tau^k$ and $\Omega_1^k > 0$. □

Lemma 5.2 *Given Assumptions 5.1 and a type one point w^k, if $\alpha_c^k = \beta_1^k = \alpha_\Delta^k$, then*

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2 \Delta^k.$$

Proof: Since $\alpha_0^{\min} \notin (0, \beta_1^k)$, then either $\alpha_0^{\min} \leq 0$ or $\alpha_0^{\min} \geq \alpha_\Delta^k$.

- $\alpha_0^{\min} \leq 0 \Rightarrow \Omega_1^k \leq 0$ and $\frac{1}{2}(\alpha_c^k)^2\|(Z^k)^T\gamma^k\|_2^2\Omega_1^k \leq 0$. Therefore,

$$\begin{aligned}\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) &= \alpha_c^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}(\alpha_c^k)^2\|(Z^k)^T\gamma^k\|_2^2\Omega_1^k \\ &\geq \alpha_c^k\|(Z^k)^T\gamma^k\|_2^2 \\ &= \frac{\Delta^k\|(Z^k)^T\gamma^k\|_2^2}{\|d_c^k\|_\infty}.\end{aligned}$$

Since $\|d_c^k\|_\infty \leq \|d_c^k\|_2 = \|(Z^k)^T\gamma^k\|_2$, it follows that

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \|(Z^k)^T\gamma^k\|_2\Delta^k > \frac{1}{2}\|(Z^k)^T\gamma^k\|_2\Delta^k.$$

- $\alpha_0^{\min} \geq \alpha_\Delta^k \Rightarrow \alpha_\Delta^k\Omega_1^k \leq 1$. Therefore,

$$\begin{aligned}\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) &= \alpha_c^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}(\alpha_c^k)^2\|(Z^k)^T\gamma^k\|_2^2\Omega_1^k \\ &= \alpha_\Delta^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}\alpha_\Delta^k\|(Z^k)^T\gamma^k\|_2^2(\alpha_\Delta^k\Omega_1^k) \\ &\geq \alpha_\Delta^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}\alpha_\Delta^k\|(Z^k)^T\gamma^k\|_2 \\ &= \frac{1}{2}\alpha_\Delta^k\|(Z^k)^T\gamma^k\|_2^2 \\ &= \frac{\Delta^k\|(Z^k)^T\gamma^k\|_2^2}{2\|d_c^k\|_\infty} \\ &\geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2\Delta^k. \qquad \square\end{aligned}$$

Lemma 5.3 *Given Assumptions 5.1 and a type one point, w^k, if $\alpha_c^k \geq \beta_1^k$, then*

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2^2\beta_1^k.$$

Proof: Since $\alpha_c^k \geq \beta_1^k$, $\varphi^k(\alpha_c^k d_c^k) \leq \varphi^k(\beta_1^k d_c^k)$. Therefore,

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \varphi^k(0) - \varphi^k(\beta_1^k d_c^k) = \beta_1^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}(\beta_1^k)^2\|(Z^k)^T\gamma^k\|_2^2\Omega_1^k.$$

Consider the two cases for Ω_1^k.

- If $\Omega_1^k \leq 0$, then $(\beta_1^k)^2\|(Z^k)^T\gamma^k\|_2^2\Omega_1^k < 0$, and

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \beta_1^k\|(Z^k)^T\gamma^k\|_2^2 > \frac{1}{2}\|(Z^k)^T\gamma^k\|_2^2\beta_1^k.$$

- If $\Omega_1^k > 0$, then $\alpha_0^{\min} \geq \beta_1^k$, $\beta_1^k\Omega_1^k \leq 1$, and

$$\begin{aligned}\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) &\geq \beta_1^k\|(Z^k)^T\gamma^k\|_2^2 - \frac{1}{2}\beta_1^k\|(Z^k)^T\gamma^k\|_2^2(\Omega_1^k\beta_1^k) \\ &\geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2^2\beta_1^k. \qquad \Box\end{aligned}$$

The above three results are combined into a single observation and used to establish several other bounds.

Lemma 5.4 *Given Assumptions 5.1 and a type one point* w^k*,*

$$\varphi^k(0) - \varphi^k(\alpha_c^k d_c^k) \geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2 \min\left(\frac{\|(Z^k)^T\gamma^k\|_2}{\tau^k}, \Delta^k, \|(Z^k)^T\gamma^k\|_2\beta_1^k\right).$$

Proof: The conditions of one of the Lemmas 5.1, 5.2, and 5.3 must be satisfied for any type one point and direction d_c^k. Therefore, these three results can be combined. $\Box$

Corollary 5.5 *Given Assumptions 5.1 and a type one point* w^k*, the approximate trust region direction* d^k *defined at* w^k *satisfies*

$$\varphi^k(0) - \varphi^k(d^k) \geq \frac{1}{2}\|(Z^k)^T\gamma^k\|_2 \min\left(\frac{\|(Z^k)^T\gamma^k\|_2}{\tau^k}, \Delta^k, \|(Z^k)^T\gamma^k\|_2\beta_1^k\right).$$

Proof: The result follows immediately from the previous lemma and the fact that d^k is accepted as the approximate trust region solution at w^k only if $\varphi^k(d^k) \leq \varphi^k(\alpha_c^k d_c^k)$. $\Box$

Corollary 5.6 *Given Assumptions 5.1 and a type one point* w^k*, if* d^k *is a successful trust region direction, then*

$$p_\mu(w^k) - p_\mu(w^k + d^k) \geq \frac{1}{2}b_1\|(Z^k)^T\gamma^k\|_2 \min\left(\frac{\|(Z^k)^T\gamma^k\|_2}{\tau^k}, \Delta^k, \|(Z^k)^T\gamma^k\|_2\beta_1^k\right).$$

Proof: The result follows immediately from Corollary 5.5 and the fact that d^k is a successful descent direction only if

$$p_\mu(w^k) - p_\mu(w^k + d^k) \geq b_1(\varphi^k(0) - \varphi^k(d^k)). \qquad \square$$

5.3.2 Further on the Penalty Function Decrease

In this section, the lower bound established in Corollary 5.6 is shown to be bounded away from zero. To prove this result, bounds are first established on β^k, the smallest breakpoint along d_c^k, on the absolute difference between the value of the model function at d^k and the penalty function value at $w^k + d^k$, and, finally, on the size of the trust region radius over a sequence of type one points.

A lower bound on β_1^k is established first.

Lemma 5.7 *Given Assumptions 5.1 and a type one point, the first breakpoint β_1^k along the Cauchy direction d_c^k before the trust region boundary is encountered is bounded away from zero.*

Proof: Assume the contrary, that there exists a subsequence of iterations $\{k_i\}$ such that

$$\liminf_{i\to\infty} \beta_1^{k_i} = 0. \tag{5.1}$$

For each iteration k_i, $\beta_1^{k_i}$ is the point along d^{k_i} at which some ϵ-inactive penalty term changes shape, that is, there exist values u^{k_i}, v^{k_i} and z^{k_i} such that

$$u^{k_i} + \beta v^{k_i} + \beta^2 z^{k_i} = 0,$$

where

- u^{k_i} is the value of an ϵ-inactive penalty term (i.e. $|u^{k_i}| > \epsilon$),

- v^{k_i} is the rate of the first order change along the direction $d_c^{k_i}$, and
- z^{k_i} is the rate of the second order change along the direction $d_c^{k_i}$.

Note that by Assumptions 5.1, there exists $M_v > 0$ such that $|v^{k_i}| \leq M_v$ and $M_z > 0$ such that $|z^{k_i}| \leq M_z$.

Consider the following cases, on any iteration k_i.

- if $z^{k_i} = 0$ then $\beta_1^{k_i} > 0$ satisfies $u^{k_i} + \beta_1^{k_i} v^{k_i} = 0$, i.e.

$$\beta_1^{k_i} = -u^{k_i}/v^{k_i} = |u^{k_i}|/|v^{k_i}| > \epsilon/M_v.$$

- if $v^{k_i} = 0$ then $\beta_1^{k_i} > 0$ satisfies $u^{k_i} + \beta^2 z^{k_i} = 0$, i.e.

$$\beta_1^{k_i} = \sqrt{-u^{k_i}/z^{k_i}} = \sqrt{|u^{k_i}|/|z^{k_i}|} > \sqrt{\epsilon/M_z}.$$

- if $v^{k_i} \neq 0$, $z^{k_i} \neq 0$, and $\beta_1^{k_i}$ is positive and real, if $(v^{k_i})^2 - 4u^{k_i} z^{k_i} = 0$, then

$$\beta_1^{k_i} = -\frac{v_{k_i}}{2z^{k_i}} = \frac{|v^{k_i}|}{2|z^{k_i}|} = \frac{\sqrt{4|u^{k_i}||z^{k_i}|}}{2|z^{k_i}|} = \frac{\sqrt{|u^{k_i}|}}{\sqrt{|z^{k_i}|}} > \sqrt{\frac{\epsilon}{M_z}}.$$

When $\beta_1^{k_i}$ corresponds to any of the above cases, for specific indices k_i or in the limit, then $\beta_1^{k_i}$ is bounded below by the constant $\min(\epsilon/M_v, \sqrt{\epsilon/M_z})$. Because of our assumption, the iterations in the subsequence $\{k_i\}$ cannot belong to any of the cases above. Therefore, if $\beta_1^{k_i}$ is positive and real, then

$$v^{k_i} \neq 0, \ z^{k_i} \neq 0 \text{ and } (v^{k_i})^2 - 4u^{k_i} z^{k_i} > 0,$$

for all finite values of k_i and in the limit as $i \to \infty$.

Since

$$\liminf_{i \to \infty} \beta_1^{k_i} = 0,$$

it follows that

$$\liminf_{i\to\infty} \frac{-v^{k_i} \pm \sqrt{(v^{k_i})^2 - 4u^{k_i}z^{k_i}}}{2z^{k_i}} = 0.$$

Because $|z^{k_i}| \leq M_z$, it follows that

$$\liminf_{i\to\infty} \left(-v^{k_i} \pm \sqrt{(v^{k_i})^2 - 4u^{k_i}z^{k_i}}\right) = 0.$$

Following along, we necessarily have that

$$\begin{aligned} & \liminf_{i\to\infty} -v^{k_i} = \liminf_{i\to\infty} \pm\sqrt{(v^{k_i})^2 - 4u^{k_i}z^{k_i}} \\ \Leftrightarrow \quad & \liminf_{i\to\infty} (v^{k_i})^2 = \liminf_{i\to\infty} \left((v^{k_i})^2 - 4u^{k_i}z^{k_i}\right) \\ \Leftrightarrow \quad & \liminf_{i\to\infty} 4u^{k_i}z^{k_i} = 0. \end{aligned}$$

Recall that $|u^{k_i}| > \epsilon$ for all k_i. Since $z^{k_i} = 0$ implies $\beta_1^{k_i} > \epsilon/M_v$, this contradicts (5.1). Therefore, there exists a constant δ_β such that

$$\beta_1^k \geq \delta_\beta$$

on all iterations. □

Next, an upper bound on the absolute difference between the penalty function value at the point $w^k + d^k$, for a trust region direction d^k, and the model function value $\varphi^k(d^k)$ is developed.

Lemma 5.8 *Given Assumptions 5.1 and an approximate trust region solution d^k at a type one point w^k, there exists a constant $q_0 > 0$ such that*

$$|p_\mu(w^k + d^k) - \varphi^k(d^k)| \leq q_0(\Delta^k)^2\tau^k.$$

Proof: This result is inspired by Lemma 3.9 in [37]. Assume that $\|d^k\|$ is small enough that the following conditions are all satisfied.

$$\begin{array}{ll} \text{for } i \in T'(\epsilon): & \text{neg}[G_i(w^k + d^k)] = \text{neg}[G_i(w^k)] \\ \text{for } i \in P'_\lambda(\epsilon) \cup P^0_\lambda(\epsilon): & \lambda_i(w^k + d^k) < g_i(w^k + d^k) \\ \text{for } i \in P'_g(\epsilon) \cup P^0_g(\epsilon): & g_i(w^k + d^k) < \lambda_i(w^k + d^k) \\ \text{for } i \in P'_g(\epsilon): & \text{sign}[g_i(w^k + d^k)] = \text{sign}[g_i(w^k)] \\ \text{for } i \in P'_=(\epsilon): & \text{sign}[\min(\lambda_i(w^k + d^k), g_i(w^k + d^k))] = \text{sign}[\lambda_i(w^k)] \\ \text{for } i \in M'(\epsilon): & \text{sign}[c_i(w^k + d^k)] = \text{sign}[c_i(w^k)]. \end{array}$$

From Lemma 4.1, we can write

$$\begin{aligned} \varphi^k(d^k) &= \varphi^k(0) + (d^k)^T\gamma^k + \frac{1}{2}(d^k)^T B^k d^k \\ &= p_\mu(w^k) + (d^k)^T\gamma^k + \frac{1}{2}\|d^k\|_2^2\Omega^k(d^k). \end{aligned}$$

Also, since d^k maintains all the current ϵ-activities up to first order,

$$\lambda_i(w^k + d^k) = \lambda_i(w^k) + (d^k)^T\nabla\lambda_i(w^k) = \lambda_i(w^k), \text{ for } i \in P^\epsilon_\lambda.$$

The following notation shall be used.

- For $i \in T'(\epsilon)$, $\rho_i = \text{neg}[G_i(w^k)]$.
- For $i \in T^0(\epsilon)$, $\rho_i^+ = \text{neg}[G_i(w^k + d^k)]$.
- For $i \in P'_\lambda(\epsilon)$, $\sigma_i = \text{sign}[\lambda_i(w^k)]$.
- For $i \in P'_g(\epsilon)$, $\sigma_i = \text{sign}[g_i(w^k)]$.
- For $i \in P^0_g(\epsilon)$, $\sigma_i^+ = \text{sign}[g_i(w^k + d^k)]$.
- For $i \in P'_=(\epsilon)$, $\sigma_i = \text{sign}[\lambda_i(w^k)]$ and $\rho_i^+ = \text{neg}[g_i(w^k + d^k) - \lambda_i(w^k + d^k)]$.
- For $i \in P^0_=(\epsilon)$,

  $$\sigma_i = \text{sign}[\min(\lambda_i(w^k), g_i(w^k))], \text{ and } \sigma_i^+ = \text{sign}[\min(\lambda_i(w^k + d^k), g_i(w^k + d^k))],$$

$$\rho_i = \text{neg}[g_i(w^k) - \lambda_i(w^k)], \text{ and } \rho_i^+ = \text{neg}[g_i(w^k + d^k) - \lambda_i(w^k + d^k)].$$

- For $i \in M'(\epsilon)$, $\sigma_i = \text{sign}[c_i(w^k)]$.

- For $i \in M^0(\epsilon)$, $\sigma_i^+ = \text{sign}[c_i(w^k + d^k)]$.

Using this notation,

$$\begin{aligned}
p_\mu(w^k + d^k) &= F(w^k) + (d^k)^T \nabla F(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(F, w^k, d^k) \\
&\quad -\mu \sum_{T'(\epsilon)} \rho_i (G_i(w^k) + (d^k)^T \nabla G_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(G_i, w^k, d^k)) \\
&\quad -\mu \sum_{T^0(\epsilon)} \rho_i^+ (G_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(G_i, w^k, d^k)) \\
&\quad +\mu \sum_{P'_\lambda(\epsilon)} \sigma_i (\lambda_i(w^k) + (d^k)^T \nabla \lambda_i(w^k)) + \mu \sum_{P^0_\lambda(\epsilon)} |\lambda_i(w^k)| \\
&\quad +\mu \sum_{P'_g(\epsilon)} \sigma_i (g_i(w^k) + (d^k)^T \nabla g_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(g_i, w^k, d^k)) \\
&\quad +\mu \sum_{P^0_g(\epsilon)} \sigma_i^+ (g_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(g_i, w^k, d^k)) \\
&\quad +\mu \sum_{P'_=(\epsilon)} \sigma_i (\lambda_i(w^k) + (d^k)^T \nabla \lambda_i(w^k) \\
&\qquad\qquad + \rho_i^+ (g_i(w^k) - \lambda_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(g_i, w^k, d^k))) \\
&\quad +\mu \sum_{P^0_=(\epsilon)} |\min(\lambda_i(w^k), g_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(g_i, w^k, d^k))| \\
&\quad +\mu \sum_{M'(\epsilon)} \sigma_i (c_i(w^k) + (d^k)^T \nabla c_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(c_i, w^k, d^k)) \\
&\quad +\mu \sum_{M^0(\epsilon)} \sigma_i^+ (c_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(c_i, w^k, d^k)) \\
&= p_\mu(w^k) + (d^k)^T \gamma^k + \frac{1}{2}\|d^k\|_2^2 \Omega_p(w^k, d^k) \\
&\quad -\mu \sum_{T^\epsilon} G_i(w^k)(\rho_i^+ - \rho_i) + \mu \sum_{P_g^\epsilon} g_i(w^k)(\sigma_i^+ - \sigma_i) \\
&\quad +\mu \sum_{P'_=(\epsilon)} \sigma_i (g_i(w^k) - \lambda_i(w^k))(\rho_i^+ - \rho_i)
\end{aligned}$$

$$+\mu \sum_{P^0_{=}(\epsilon)} (\lambda_i(w^k)(\sigma_i^+ - \sigma_i) + (g_i(w^k) - \lambda_i(w^k))(\sigma_i^+\rho_i^+ - \sigma_i\rho_i))$$
$$+\mu \sum_{M^0(\epsilon)} c_i(w^k)(\sigma_i^+ - \sigma_i).$$

Therefore, it follows that

$$p_\mu(w^k + d^k) - \varphi(d^k) = \kappa_1(w^k, d^k) + \mu\kappa_2(w^k, d^k),$$

where

$$\begin{aligned}
\kappa_1(w^k, d^k) &= \frac{1}{2}\|d^k\|_2^2(\Omega_p(w^k, d^k) - \Omega^k(d^k)) \\
\kappa_2(w^k, d^k) &= -\sum_{T^0(\epsilon)} G_i(w^k)(\rho_i^+ - \rho_i) + \sum_{P^0_g(\epsilon)} g_i(w)(\sigma_i^+ - \sigma_i) \\
&\quad + \sum_{P'_{=}(\epsilon)} \sigma_i(g_i(w^k) - \lambda_i(w^k))(\rho_i^+ - \rho_i) \\
&\quad + \sum_{P^0_{=}(\epsilon)} (\lambda_i(w^k)(\sigma_i^+ - \sigma_i) + (g_i(w^k) - \lambda_i(w^k))(\sigma_i^+\rho_i^+ - \sigma_i\rho_i)) \\
&\quad + \sum_{M^0(\epsilon)} c_i(w^k)(\sigma_i^+ - \sigma_i).
\end{aligned}$$

Applying the triangle inequality,

$$|p_\mu(w^k + d^k) - \varphi(d^k)| \le |\kappa_1(w^k, d^k)| + \mu|\kappa_2(w^k, d^k)|.$$

Looking more closely at κ_1, it follows that

$$|\kappa_1(w^k, d^k)| \le \frac{1}{2}\|d^k\|_2^2(|\Omega_p(w^k, d^k)| + |\Omega^k(d^k)|) \le \frac{1}{2}\|d^k\|_2^2(L + \tau^k), \qquad (5.2)$$

where Assumption 5.1 guarantees the existence of the constant L.

The second expression κ_2 requires a more detailed analysis.

$$\begin{aligned}
|\kappa_2(w, d)| &\le \sum_{T^0(\epsilon)} |G_i(w^k)||\rho_i^+ - \rho_i| + \sum_{P^0_g(\epsilon)} |g_i(w^k)||\sigma_i^+ - \sigma_i| \\
&\quad + \sum_{P'_{=}(\epsilon)} |g_i(w^k) - \lambda_i(w^k)||\rho_i^+ - \rho_i|
\end{aligned}$$

$$+ \sum_{i \in P^0_=(\epsilon)} |(\lambda_i(w^k)(\sigma_i^+ - \sigma_i) + (g_i(w^k) - \lambda_i(w^k))(\sigma_i^+ \rho_i^+ - \sigma_i \rho_i))|$$
$$+ \sum_{M^0(\epsilon)} |c_i(w^k)||\sigma_i^+ - \sigma_i|.$$

For $i \in T^0(\epsilon)$, let $\chi_i = |G_i(w^k)||\rho_i^+ - \rho_i|$. Each $i \in T^0(\epsilon)$ must satisfy at least one of the following conditions:

1. $\rho_i^+ = \rho_i \Rightarrow \chi_i = 0$.

2. $\rho_i^+ = 1$ (or $G_i(w+d) < 0$) and $\rho_i = 0$ (or $G_i(w) \geq 0$). Since

$$G_i(w^k + d^k) = G_i(w^k) + \frac{1}{2}\|d^k\|_2^2 \Omega(G_i, w^k, d^k) < 0,$$

it follows that

$$\frac{1}{2}\|d^k\|_2^2 \Omega(G_i, w^k, d^k) < -G_i(w^k),$$

and, since the second order change is negative,

$$|G_i(w^k)| \leq \frac{1}{2}\|d^k\|_2^2 |\Omega(G_i, w^k, d^k)|.$$

3. $\rho_i^+ = 0$ (or $G_i(w^k + d^k) \geq 0$) and $\rho_i = 1$ (or $G_i(w^k) < 0$). Following a similar reasoning to that used above,

$$G_i(w^k) > -\frac{1}{2}\|d^k\|_2^2 \Omega(G_i, w^k, d^k), \text{ and } |G_i(w^k)| \leq \frac{1}{2}\|d^k\|_2^2 |\Omega(G_i, w^k, d^k)|.$$

Divide $T^0(\epsilon)$ into three distinct sets, T_1, T_2 and T_3, corresponding to the cases described above. Then,

$$\begin{aligned}
\sum_{T^0(\epsilon)} \chi_i &= \sum_{T_1} \chi_i + \sum_{T_2} \chi_i + \sum_{T_3} \chi_i \\
&= \sum_{T_2} |G_i(w^k)| + \sum_{T_3} |G_i(w^k)| \\
&\leq \frac{1}{2}\|d^k\|_2^2 (\sum_{T_2} |\Omega(G_i, w^k, d^k)| + \sum_{T_3} |\Omega(G_i, w^k, d^k)|)
\end{aligned}$$

$$\begin{aligned} &\leq \frac{1}{2}\|d^k\|_2^2 \mathrm{Size}(T_2 \cup T_3) \max |\Omega(G_i, w^k, d^k)| \\ &\leq \frac{1}{2}\|d^k\|_2^2 t L_G, \end{aligned}$$

where t is the number of upper level constraints in the bilevel problem and L_G is a constant bounding the maximum curvature of all the functions G_i. The existence of L_G is guaranteed by Assumptions 5.1.

Using similar analysis, it can be shown that

$$\begin{aligned} \sum_{P_g^0(\epsilon)} \chi_i &\leq \|d^k\|_2^2 L_g \mathrm{Size}(P_g^0(\epsilon)) \\ \sum_{P'_=(\epsilon)} \chi_i &\leq \frac{1}{2}\|d^k\|_2^2 L_g \mathrm{Size}(P'_=(\epsilon)) \\ \sum_{M^0(\epsilon)} \chi_i &\leq \|d^k\|_2^2 L_c \mathrm{Size}(M^0(\epsilon)) = \|d^k\|_2^2 L_c m, \end{aligned}$$

where L_g and L_c, giving the maximum curvature of all g_i for $i \in P$, and c_i, for $i \in M$, respectively, exist by Assumption 5.1.

Next, examine the remaining sum, over $P_=^0(\epsilon)$, more closely. Let, for $i \in P_=^0(\epsilon)$,

$$\chi_i = |\lambda_i(w^k)(\sigma_i^+ - \sigma_i) + (g_i(w^k) - \lambda_i(w^k))(\sigma_i^+ \rho_i^+ - \sigma_i \rho_i)|.$$

Consider the following cases for $i \in P_=^0(\epsilon)$.

1. If $\sigma_i^+ = \sigma_i$, then

$$\begin{aligned} \chi_i &= |\lambda_i(w^k) \cdot 0 + (g_i(w^k) - \lambda_i(w^k))\sigma_i(\rho_i^+ - \rho_i)| \\ &= |g_i(w^k) - \lambda_i(w^k)||\rho_i^+ - \rho_i|. \end{aligned}$$

   (a) If $\rho_i^+ = \rho_i$, then $\chi_i = 0$.

   (b) If $\rho_i^+ \neq \rho_i$, then

$$\chi_i = |g_i(w^k) - \lambda_i(w^k)| \leq \frac{1}{2}\|d^k\|_2^2 |\omega(g_i, w^k, d^k)|,$$

   using the same analysis as above.

2. If $\sigma_i^+ = -\sigma_i$, then

$$\begin{aligned}\chi_i &= |2\lambda_i(w^k)\sigma_i^+ + (g_i(w^k) - \lambda_i(w^k))\sigma_i^+(\rho_i^+ - \rho_i)| \\ &= |2\lambda_i(w) + (g_i(w) - \lambda_i(w))(\rho_i^+ - \rho_i)|.\end{aligned}$$

We now look further at the possibilities for this case.

(a) If $\rho_i^+ = 0$ and $\rho_i = 0$, then it follows that

$$\begin{aligned}\min(\lambda_i(w^k + d_k), g_i(w^k + d^k)) &= \lambda_i(w^k + d^k) \\ \min(\lambda_i(w^k), g_i(w^k)) &= \lambda_i(w^k),\end{aligned}$$

and $\text{sign}[\lambda_i(w^k + d^k)] = -\text{sign}[\lambda_i(w^k)]$. However, since $\lambda_i(w^k + d^k) = \lambda_i(w^k)$, this condition cannot be satisfied. Therefore, this case is empty.

(b) If $\rho_i^+ = 0$ and $\rho_i = 1$, then $\text{sign}[\lambda_i(w^k)] = -\text{sign}[g_i(w^k)]$. In this case,

$$\chi_i = |g_i(w^k) + \lambda_i(w^k)| \leq |g_i(w^k)| + |\lambda_i(w^k)| = |g_i(w^k) - \lambda_i(w^k)|,$$

since $g_i(w^k)$ and $\lambda_i(w^k)$ are of opposite signs. Since $g_i(w^k + d^k) - \lambda_i(w^k + d^k)$ and $g_i(w^k) - \lambda_i(w^k)$ are of different signs, it follows that

$$|g_i(w^k) - \lambda_i(w^k)| \leq \frac{1}{2}||d^k||_2^2|\Omega(g_i, w^k, d^k)|,$$

and

$$\chi_i \leq \frac{1}{2}||d^k||_2^2|\Omega(g_i, w^k, d^k)|.$$

(c) If $\rho_i^+ = 1$ and $\rho_i = 0$, then $\text{sign}[g_i(w^k + d^k)] = -\text{sign}[\lambda_i(w^k)]$. Again, we have that

$$\chi_i = |g_i(w^k) + \lambda_i(w^k)|..$$

Consider the two possibilities for $\text{sign}[g_i(w^k)]$.

i. $\text{sign}[g_i(w^k)] = \text{sign}[g_i(w^k + d^k)] = -\text{sign}[\lambda_i(w^k)]$: as in the above case,

$$\chi_i \le |g_i(w^k)| + |\lambda_i(w^k)| = |g_i(w^k) - \lambda_i(w^k)| \le \frac{1}{2}\|d^k\|_2^2|\Omega(g_i, w^k, d^k)|.$$

ii. $\text{sign}[g_i(w^k)] = -\text{sign}[g_i(w^k + d^k)] = \text{sign}[\lambda_i(w^k)]$: since $g_i(w^k)$ has a different sign than $g_i(w^k + d^k)$, it follows that

$$|g_i(w^k)| \le \frac{1}{2}\|d^k\|_2^2|\Omega(g_i, w^k, d^k)|.$$

Therefore,

$$\begin{aligned}
\chi_i &= |\lambda_i(w^k) - g_i(w^k) + 2g_i(w^k)| \\
&\le |\lambda_i(w^k) - g_i(w^k)| + 2|g_i(w^k)| \\
&\le \frac{1}{2}\|d^k\|_2^2|\Omega(g_i, w^k, d^k)| + \|d^k\|_2^2\Omega(g_i, w^k, d^k)| \\
&= \frac{3}{2}\|d^k\|_2^2|\Omega(g_i, w^k, d^k)|.
\end{aligned}$$

(d) If $\rho_i^+ = 1$ and $\rho_i = 1$, then $\text{sign}[g_i(w^k + d^k)] = -\text{sign}[g_i(w^k)]$ and

$$\chi_i = 2|g_i(w^k)| \le \|d^k\|_2^2|\Omega(g_i, w^k, d^k)|,$$

since $g_i(w^k + d^k)$ and $g_i(w^k)$ are of different signs.

Combining all the cases above, we get that

$$\sum_{P_=^\epsilon} \chi_i \le \frac{3}{2}\|d^k\|_2^2 L_g \text{Size}(P_=^0(\epsilon)).$$

Therefore, it follows that

$$\begin{aligned}
|\kappa_2(w^k, d^k)| &\le \frac{1}{2}\|d^k\|_2^2 L_G t + \|d^k\|_2^2 L_g \text{Size}(P_g^0(\epsilon)) + \frac{1}{2}\|d^k\|_2^2 L_g \text{Size}(P_=^\prime(\epsilon)) \\
&\quad + \frac{3}{2}\|d^k\|_2^2 L_g \text{Size}(P_=^0(\epsilon)) + \|d^k\|_2^2 L_c m \\
&\le q_1\|d^k\|_2^2,
\end{aligned}$$

where

$$q_1 = \frac{1}{2}L_G t + \frac{3}{2}pL_g + L_c m.$$

Combining this result with (5.2), and $\tau^k \geq 1$, we have

$$\begin{aligned} |p_\mu(w^k + d^k) - \varphi^k(d^k)| &\leq \|d^k\|_2^2 \left(L + \tau^k + \mu q_1\right) \\ &\leq (n+m+p)\|d^k\|_\infty^2 \left(L + \tau^k + \mu q_1\right) \\ &\leq (n+m+p)(\Delta^k)^2(L\tau^k + \tau^k + \mu q_1 \tau^k) \\ &= q_0(\Delta^k)^2\tau^k, \end{aligned}$$

where

$$q_0 = (n+m+p)(L+1+\mu q_1). \qquad \square$$

A lower bound on Δ^k can now be established, assuming that Λ satisfies the following assumption, which is similar to an assumption stated in equation (104) on page 446 in [38].

Assumption 5.3 *The algorithm parameter Λ is small enough that the following condition is satisfied:*

$$\Lambda < \min\left(2\tau^0\Delta^0, \frac{4q\tau^0\Delta^0}{1-b_2}, 2\left(\frac{q_0(\tau^0\Delta^0)^2}{1-b_2}\right)^{1/3}, \delta_\beta\right).$$

Lemma 5.9 *Given Assumptions 5.1 and 5.3, and a type one point w^k, for all iterations k, there exists a constant $\delta > 0$ such that,*

$$\Delta^k\tau^k \geq \delta.$$

Proof: This proof uses the technique of Lemma 7 (page 446) in [38]. Define δ as

$$\delta = \frac{1}{2}\Lambda \min\left(1, \frac{1-b_2}{2q_0}, \sqrt{\frac{\Lambda(1-b_2)}{2q_0}}\right). \tag{5.3}$$

Assume that the lemma is not true, i.e. that there exists an index J such that

$$\Delta^J \tau^J < \delta. \tag{5.4}$$

The definition of δ and Assumption 5.3 ensure that $\Delta^0\tau^0 \geq \delta$, so $J \geq 1$. Without loss of generality, assume that J is the smallest index satisfying (5.4), i.e. $\Delta^k\tau^k \geq \delta$ for $k = 0 : J-1$.

Define ξ^{J-1} as follows:

$$\begin{aligned} \xi^{J-1} &= \left|\frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1\right| \\ &= \left|\frac{p_\mu(w^{J-1}+d^{J-1}) - p_\mu(w^{J-1})}{\varphi^{J-1}(d^{J-1}) - \varphi^{J-1}(0)} - 1\right| \\ &= \left|\frac{p_\mu(w^{J-1}+d^{J-1}) - \varphi^{J-1}(d^{J-1})}{\varphi^{J-1}(d^{J-1}) - \varphi^{J-1}(0)}\right|, \end{aligned}$$

since $p_\mu(w^{J-1}) = \varphi^{J-1}(0)$. Using the results of Corollary 5.5 and Lemma 5.8, Assumption 5.3, and the fact that w^k is a type one point, it follows that

$$\xi^{J-1} \leq \frac{2q_0(\Delta^{J-1})^2\tau^{J-1}}{\Lambda\min(\frac{\Lambda}{\tau^{J-1}}, \Delta^{J-1}, \Lambda\beta_1^{J-1})} \leq \frac{2q_0(\Delta^{J-1})^2\tau^{J-1}}{\Lambda\min(\frac{\Lambda}{\tau^{J-1}}, \Delta^{J-1}, \Lambda^2)}.$$

Consider the following:

$$\begin{aligned} \tau^{J-1}\Delta^{J-1} &\leq \tau^J\Delta^{J-1} && \text{since } \tau^{J-1} \leq \tau^J \\ &\leq 2\tau^J\Delta^J && \text{since } \Delta^{J-1} \leq 2\Delta^J \text{ even if iteration is unsuccessful} \\ &< 2\delta && \text{from (5.4)} \\ &\leq \Lambda && \text{since } \delta \leq \Lambda/2 \text{ from (5.3).} \end{aligned}$$

Since $\Lambda/\tau^{J-1} \geq \Delta^{J-1}$,

$$\xi^{J-1} \leq \frac{2q_0(\Delta^{J-1})^2\tau^{J-1}}{\Lambda\min(\Delta^{J-1}, \Lambda^2)} \leq \frac{4q_0\Delta^{J-1}\delta}{\Lambda\min(\Delta^{J-1}, \Lambda^2)}.$$

Consider the two possibilities for the min term separately.

- $\Delta^{J-1} \leq \Lambda^2$:

$$\xi^{J-1} \leq \frac{4q_0\delta}{\Lambda} \leq \frac{4q_0}{\Lambda}\frac{\Lambda(1-b_2)}{4q_0} = 1 - b_2.$$

- $\Lambda^2 < \Delta^{J-1}$:

$$\xi^{J-1} \leq \frac{4q_0\Delta^{J-1}\delta}{\Lambda^3} \leq \frac{4q_0\Delta^{J-1}\tau^{J-1}\delta}{\Lambda^3} \leq \frac{8q_0\delta^2}{\Lambda^3} \leq \frac{8q_0}{\Lambda^3}\frac{\Lambda^3(1-b_2)}{8q_0} = 1 - b_2.$$

Therefore, in either case, $\xi^{J-1} \leq 1 - b_2$, or

$$\left|\frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1\right| \leq 1 - b_2 \Rightarrow \frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1 \geq b_2 - 1 \Rightarrow \frac{\chi_a^{J-1}}{\chi_p^{J-1}} \geq b_2,$$

i.e. iteration $J-1$ is very successful and $\Delta^{J-1} \leq \Delta^J$. So,

$$\Delta^{J-1}\tau^{J-1} \leq \Delta^J\tau^{J-1} \leq \Delta^J\tau^J < \delta,$$

which contradicts the fact that J is the smallest index satisfying (5.4). The result follows immediately from the contradiction. □

The results just established are now used to provide more information about the decrease in p_μ from a type one point.

Lemma 5.10 *Given Assumptions 5.1, 5.2, and 5.3, if, for all k, w^k is a type one point, then for any successful trust region direction d^k, there exists a positive constant δ_p such that*

$$p_\mu(w^k + d^k) - p_\mu(w^k) \geq \delta_p.$$

Proof: It follows from Corollary 5.5, Lemma 5.9, and w^k, a type one point, that

$$\begin{aligned} p_\mu(w^k + d^k) - p_\mu(w^k) &\geq \frac{1}{2}b_1\|(Z^k)^T\gamma^k\|_2 \min\left(\frac{\|(Z^k)^T\gamma^2\|_2}{\tau^k}, \Delta^k, \|(Z^k)^T\gamma^k\|_2\beta_1^k\right) \\ &\geq \frac{1}{2}b_1\Lambda \min\left(\frac{\Lambda}{\tau^k}, \Delta^k, \Lambda\beta_1^k\right) \\ &\geq \frac{1}{2}b_1\Lambda \min\left(\frac{\Lambda}{\tau^k}, \frac{\delta}{\tau^k}, \Lambda\beta_1^k\right). \end{aligned}$$

By Lemma 5.7, $\beta_1^k \geq \delta_\beta$, so

$$p_\mu(w^k + d^k) - p_\mu(w^k) \geq \frac{1}{2} b_1 \Lambda \min\left(\frac{\Lambda}{\tau^k}, \frac{\delta}{\tau^k}, \Lambda\delta_\beta\right).$$

From Assumption 5.2, τ^k is bounded above by a constant $\tau^{\max}$, so

$$p_\mu(w^k + d^k) - p_\mu(w^k) \geq \frac{1}{2} b_1 \Lambda \min\left(\frac{\Lambda}{\tau^{\max}}, \frac{\delta}{\tau^{\max}}, \Lambda\delta_\beta\right) = \delta_p. \qquad \square$$

If iteration k is a successful iteration from a type one point w^k, the decrease in the penalty function along the trust region direction is bounded away from zero. If all the iterations correspond to type one points, only a finite number of successful iterations are possible, since, by Assumption 5.1, p_μ is bounded below. This result is proved formally in the next section.

5.3.3 Approaching a Near Stationary Point

Before proving that the iterates approach a near stationary point, further notation and intermediate results are required. We first show that there exists at least one subsequence of iterates which approaches a near stationary point. After two technical results are developed for the algorithm, it is established that if there are only a finite number of successful iterations, the algorithm must approach a near stationary point. A similar result is then established for an infinite number of successful iterations, providing the desired result.

Definition 5.3 *Define the following sets:*

$$\begin{aligned} \mathcal{S} &= \{k : \text{ iteration } k \text{ is successful}\} \\ \overline{\mathcal{S}} &= \{k : \text{ iteration } k \text{ is unsuccessful}\}. \end{aligned}$$

Lemma 5.11 *Given Assumptions 5.1, 5.2, and 5.3, if, for all k, w^k is a type one point, then*

$$\liminf_{k\to\infty} \|Z_k^T \gamma_k\|_2 \le \Lambda.$$

Proof: The proof uses the technique of Theorem 8 (page 447) in [38] and proceeds by contradiction.

Assume that the result does not hold and $\|Z_k^T \gamma_k\|_2 > \Lambda$ for all iterations k.

On a successful iteration $k \in \mathcal{S}$, we have, from Corollary 5.6, that

$$p_\mu(w^k) - p_\mu(w^{k+1}) \ge \frac{1}{2} b_1 \|(Z^k)^T \gamma^k\|_2 \min(\|(Z^k)^T \gamma^k\|_2/\tau^k, \Delta^k, \|(Z^k)^T \gamma^k\|_2 \beta_k^1),$$

where $w_{k+1} = w_k + d_k$. From Assumption 5.1, p_μ is bounded below,

$$\lim_{k\to\infty} (p_\mu(w^k) - p_\mu(w^{k+1})) = 0$$

and since $p_\mu(w^k) - p_\mu(w^{k+1}) \ge 0$ for all k,

$$\sum_{k\in\mathcal{S}} (p_\mu(w^k) - p_\mu(w^{k+1})) < +\infty.$$

Therefore, using Assumption 5.3, Lemma 5.9, and $\tau^k \ge 1$, we have

$$\begin{aligned}
\sum_{k\in\mathcal{S}} (p_\mu(w^k) - p_\mu(w^{k+1})) &\ge \frac{1}{2} b_1 \sum_{k\in\mathcal{S}} \Lambda \min(\Lambda/\tau^k, \Delta^k, \Lambda\beta_1^k) \\
&\ge \frac{1}{2} b_1 \Lambda \sum_{k\in\mathcal{S}} \min(\Lambda/\tau^k, \Delta^k, \Lambda^2) \\
&\ge \frac{1}{2} b_1 \Lambda \sum_{k\in\mathcal{S}} \min(\Lambda/\tau^k, \delta/\tau^k, \Lambda^2/\tau^k) \\
&\ge \frac{1}{2} b_1 \Lambda \min(\Lambda, \delta, \Lambda^2) \sum_{k\in\mathcal{S}} 1/\tau^k.
\end{aligned}$$

Therefore, $\sum_{k\in\mathcal{S}} 1/\tau^k < +\infty$.

Let $\gamma_2 \in (0,1)$ and $\gamma_3 \ge 1$ be the modification constants for the trust region radius, i.e.

$$\text{for } k \in \overline{\mathcal{S}} : \Delta_{k+1} \le \gamma_2 \Delta_k \text{ and for } k \in \mathcal{S} : \Delta_{k+1} \le \gamma_3 \Delta_k,$$

and let p, a positive integer, satisfy $\gamma_3\gamma_2^{p-1} < 1$. In Algorithm 4.1, $\gamma_2 = \frac{1}{2}$ and $\gamma_3 = 2$, and any $p \geq 3$ is suitable.

Let $S_k = |\mathcal{S} \cap \{0, \cdots, k-1\}|$ be the number of successful iterations in the first k iterations. Define the following, mutually disjoint sets which span all the iteration indices,

$$J_1 = \{k : k \leq pS_k\} \text{ and } J_2 = \{k : k > pS_k\}.$$

We will examine the sums $\sum_{i \in J} 1/\tau_k$ for $J = J_1$ and $J = J_2$.

First, consider the sum $\sum_{i \in J_1} 1/\tau_k$, and define the two subsequences of indices:

- $\mathcal{K}^1$, the indices of J_1 in increasing order,
- $\mathcal{K}^2$, the indices of $\mathcal{S}$ in increasing order, with each index repeated p times.

Note that the jth components of the two sequences satisfy $\mathcal{K}_j^1 \geq \mathcal{K}_j^2$ and that $\mathcal{K}^2$ has at least as many components as $\mathcal{K}^1$. Therefore, $\tau^{\mathcal{K}_j^1} \geq \tau^{\mathcal{K}_j^2}$. Looking at the sums,

$$\sum_{k \in J_1} 1/\tau^k = \sum_{k \in \mathcal{K}^1} 1/\tau^k \leq \sum_{k \in \mathcal{K}^2} 1/\tau^k = p \sum_{k \in \mathcal{S}} 1/\tau^k < +\infty.$$

Next, consider the sum $\sum_{k \in J_2} 1/\tau_k$. First, for any $k \in J_2$,

$$\begin{aligned}
\Delta^k &\leq \begin{cases} \gamma_2 \Delta^{k-1} & \text{if } k-1 \in \overline{\mathcal{S}} \\ \gamma_3 \Delta^{k-1} & \text{if } k-1 \in \mathcal{S} \end{cases} \\
&\leq \begin{cases} \gamma_2^2 \Delta^{k-2} & \text{if } k-1, k-2 \in \overline{\mathcal{S}} \\ \gamma_3^2 \Delta^{k-2} & \text{if } k-1, k-2 \in \mathcal{S} \\ \gamma_2\gamma_3 \Delta^{k-2} & \text{otherwise} \end{cases} \\
&\;\;\vdots \\
&\leq \gamma_2^{k-S_k} \gamma_3^{S_k} \Delta^0 \\
&\leq \gamma_2^{k-k/p} \gamma_3^{k/p} \Delta^0 \\
&= (\gamma_3 \gamma_2^{p-1})^{k/p} \Delta^0,
\end{aligned}$$

since $S_k < k/p$, $\gamma_2 < 1$, and $\gamma_3 \geq 1$. Therefore, using Lemma 5.9,

$$1/\tau^k \leq \Delta^k/\delta \leq (\gamma_3\gamma_2^{p-1})^{k/p}\Delta^0/\delta.$$

Since $\gamma_3\gamma_2^{p-1} < 1$,

$$\sum_{k\in J_2} 1/\tau^k \leq \frac{\Delta_0}{\delta} \sum_{k\in J_2} (\gamma_3\gamma_2^{p-1})^{k/p} < +\infty.$$

Combining the information about the two sums,

$$\sum_{k=1}^{\infty} 1/\tau^k = \sum_{k\in J_1} 1/\tau^k + \sum_{k\in J_2} 1/\tau^k < +\infty,$$

which contradicts Assumption 5.2. Therefore, the result is proven. □

We have just established that there exists a subsequence of the iterates which converges to a near stationary point. To establish that all iterates will converge to such a point, it is necessary to introduce two other results.

Lemma 5.12 *Given Assumptions 5.1 and $\{q_j\}$ an infinite subsequence of the iterates, if there exists $\hat{w}$ such that*

$$\lim_{j\to\infty} w^{q_j} = \hat{w},$$

then

$$\lim_{j\to\infty} \|(Z^{q_j})^T\gamma^{q_j}\|_2 = \|\hat{Z}^T\hat{\gamma}\|_2,$$

where $\hat{Z} = Z(\hat{w})$, and $\hat{\gamma} = \gamma(\hat{w})$.

Proof: Assume for convenience and without loss of generality that each ϵ-active penalty term (with value u_i) satisfies $|u_i| < \epsilon$ or $|u_i| > \epsilon$, that is, that none of the activities have magnitude of exactly ϵ. Note that if there exists i such that $|u_i| = \epsilon$, then ϵ can be slightly increased without affecting the division of the penalty terms into ϵ-active and ϵ-inactive terms.

By the continuity of the underlying functions comprising the penalty function, we have that

$$\begin{array}{llll}
\lim_{j\to\infty} G_i(w^{q_j}) = G_i(\hat{w}) & \text{and} & \lim_{j\to\infty} \nabla G_i(w^{q_j}) = \nabla G_i(\hat{w}) & \text{for } i \in T \\
\lim_{j\to\infty} g_i(w^{q_j}) = g_i(\hat{w}) & \text{and} & \lim_{j\to\infty} \nabla g_i(w^{q_j}) = \nabla g_i(\hat{w}) & \text{for } i \in P \\
\lim_{j\to\infty} \lambda_i(w^{q_j}) = \lambda_i(\hat{w}) & \text{and} & \lim_{j\to\infty} \nabla \lambda_i(w^{q_j}) = \nabla \lambda_i(\hat{w}) & \text{for } i \in P \\
\lim_{j\to\infty} c_i(w^{q_j}) = c_i(\hat{w}) & \text{and} & \lim_{j\to\infty} \nabla c_i(w^{q_j}) = \nabla c_i(\hat{w}) & \text{for } i \in M.
\end{array}$$

Therefore, it follows that there exists an integer $J > 0$ such that for all $j > J$, all the following conditions are satisfied.

- For $i \in T^0(\epsilon)$, $|G_i(w^{q_j})| < \epsilon$ and for $i \in T'(\epsilon)$, $|G_i(w^{q_j})| > \epsilon$,
- For $i \in P_\lambda$, $\lambda_i(w^{q_j}) < g_i(w^{q_j})$. Then for $i \in P_\lambda^0(\epsilon)$, $|\lambda_i(w^{q_j})| < \epsilon$ and for $i \in P_\lambda(\epsilon)$, $|\lambda_i(w^{q_j})| > \epsilon$.
- For $i \in P_g$, $g_i(w^{q_j}) < \lambda_i(w^{q_j})$. Then for $i \in P_g^0(\epsilon)$, $|g_i(w^{q_j})| < \epsilon$ and for $i \in P_g(\epsilon)$, $|g_i(w^{q_j})| > \epsilon$.
- For $i \in P_=(\epsilon)$, $|g_i(w^{q_j}) - \lambda_i(w^{q_j})| < \epsilon$. Then, for $i \in P_=^0(\epsilon)$, $|\lambda_i(w^{q_j})| < \epsilon$ or $|g_i(w^{q_j})| < \epsilon$, and for $i \in P_='(\epsilon)$, $|\lambda_i(w^{q_j})| > \epsilon$ and $|g_i(w^{q_j})| > \epsilon$.
- For $i \in M^0(\epsilon)$, $|c_i(w^{q_j})| < \epsilon$ and for $i \in M'(\epsilon)$, $|c_i(w^{q_j})| > \epsilon$.

Consequently, for $j > J$, the ϵ-active and ϵ-inactive sets of penalty terms are correctly identified as those at $\hat{w}$. Combining this result with gradient limits defined above, it follows that

$$\lim_{j\to\infty} \gamma^{q_j} = \hat{\gamma}.$$

It similarly follows that

$$\lim_{j\to\infty} \mathcal{A}^{q_j} = \hat{\mathcal{A}} \text{ and } \lim_{j\to\infty} Z^{q_j} = \hat{Z}.$$

Combining these results, we get that

$$\lim_{j\to\infty} \|(Z^{q_j})^T\gamma^{q_j}\|_2 = \|\hat{Z}^T\hat{\gamma}\|_2$$

and the result is proven. □

Corollary 5.13 *Given Assumptions 5.1, two infinite sequences* $\{q_j\}$ *and* $\{r_j\}$*, and*

$$\lim_{j\to\infty} \|w^{q_j} - w^{r_j}\| = 0,$$

then

$$\lim_{j\to\infty} \|(Z^{q_j})^T\gamma^{q_j} - (Z^{r_j})^T\gamma^{r_j}\| = 0.$$

Proof: From the given information, there exists $\hat{w}$ such that

$$\lim_{j\to\infty} w^{q_j} = \hat{w} = \lim_{j\to\infty} w^{r_j}.$$

Therefore, from the previous result,

$$\lim_{j\to\infty} \|(Z^{q_j})^T\gamma^{q_j}\|_2 = \|\hat{Z}^T\hat{\gamma}\|_2 = \lim_{j\to\infty} \|(Z^{r_j})^T\gamma^{r_j}\|_2,$$

and the result follows immediately. □

Lemma 5.14 *Given Assumptions 5.1 and 5.2, if* S *is a finite set, then there exists* $\hat{w}$ *such that*

$$\lim_{k\to\infty} w^k = \hat{w},$$

and $\|\hat{Z}^T\hat{\gamma}\|_2 \le \Lambda$.

Proof: Since S is finite, there exists an index $K > 0$ such that all iterations $k \ge K$ must be unsuccessful, i.e.

$$w^K = w^{K+1} = w^{K+2} = \cdots,$$

and

$$\lim_{k\to\infty} w^k = w^K.$$

Since Δ^k is reduced on unsuccessful iterations, it also follows that

$$\lim_{k\to\infty} \Delta^k \;=\; 0. \tag{5.5}$$

Assume that w^K is a type one point, i.e. $\|(Z^K)^T\gamma^K\|_2 \geq \Lambda$. Therefore, from Lemma 5.9, we know that $\Delta^k\tau^K \geq \delta$, for some constant δ. Since τ^k is bounded above and below, Δ^k is bounded away from zero, which contradicts (5.5) above. Therefore, $\hat{w}$ is not a type one point, so $\|\hat{Z}^T\hat{\gamma}\|_2 \leq \Lambda$. □

Finally, it is now shown that the sequence of iterates defined by Algorithm 4.1 converge to a near stationary point.

Lemma 5.15 *Given Assumptions 5.1, 5.2, and 5.3, if S is an infinite set, then*

$$\lim_{k\in S} \|(Z^k)^T\gamma^k\|_2 \leq \Lambda.$$

Proof: This proof uses the technique of Lemma 3.15 (page 182) in [37], and proceeds by contradiction. Assume that the result is not true, i.e. there exists a subsequence of successful iterates, denoted $\{q_j\}$, for which

$$\|(Z^{q_j})^T\gamma^{q_j}\|_2 > \Lambda$$

is satisfied by the entire subsequence.

However, from Lemma 5.11, there exists a subsequence of iterates, denoted $\{r_j\}$, for which

$$\lim_{j\to\infty} \|(Z^{r_j})^T\gamma^{r_j}\|_2 \leq \Lambda.$$

Therefore, for large enough values of j,

$$\begin{aligned} \|(Z^k)^T\gamma^k\|_2 &> \Lambda \quad k = q_j, q_j+1, \cdots, r_j - 1 \\ \|(Z^{r_j})^T\gamma^{r_j}\|_2 &\leq \Lambda. \end{aligned}$$

Note that r_j is the smallest index greater than q_j for which the above conditions are satisfied.

Consider the set $\mathcal{K}_j = \{k \in \mathcal{S} : q_j \leq k \leq r_j - 1\}$. Since $q_j \in \mathcal{K}_j$, the set is nonempty. Recall, from Corollary 5.6, that for $k \in \mathcal{K}_j$ with j large enough and by Assumption 5.3,

$$\begin{aligned} p_\mu(w^k) - p_\mu(w^{k+1}) &\geq \frac{1}{2}b_1\|(Z^k)^T\gamma^k\|_2 \min(\|(Z^k)^T\gamma^k\|_2/\tau^k, \Delta^k, \|(Z^k)^T\gamma^k\|_2\beta_1^k) \\ &\geq \frac{1}{2}b_1\Lambda \min(\Lambda/\tau^k, \Delta^k, \Lambda^2). \end{aligned}$$

Since p_μ is bounded below by Assumption 5.1,

$$\lim_{k\to\infty}(p_\mu(w^k) - p_\mu(w^{k+1})) = 0,$$

and, therefore,

$$\lim_{k\in\mathcal{K}_j, j, k\to\infty} \min(\Lambda/\tau^k, \Delta^k, \Lambda^2) = 0.$$

Since τ^k is bounded above and Λ^2 is a constant, for j and k large enough, Δ^k must be the minimum term. Therefore,

$$\Delta^k \leq \frac{2}{b_1\Lambda}(p_\mu(w^k) - p_\mu(w^{k+1})).$$

Now,

$$\begin{aligned} \|w^{q_j} - w^{r_j}\|_\infty &= \|w^{q_j} - \sum_{k=q_j+1}^{r_j-1} w^k + \sum_{k=q_j+1}^{r_j-1} w^k - w^{r_j}\|_\infty \\ &\leq \sum_{k=q_j}^{r_j-1} \|w^k - w^{k+1}\|_\infty \\ &\leq \sum_{k\in\mathcal{K}_j} \|d^k\|_\infty \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{k \in \mathcal{K}_j} \Delta^k \\
&\leq \frac{2}{b_1 \Lambda} \sum_{k \in \mathcal{K}_j} (p_\mu(w^k) - p_\mu(w^{k+1})) \\
&= \frac{2}{b_1 \Lambda} (p_\mu(w^{q_j}) - p_\mu(w^{r_j})).
\end{aligned}
$$

Since

$$\lim_{j \to \infty} (p_\mu(w^{q_j}) - p_\mu(w^{r_j})) = 0,$$

it follows that

$$\lim_{j \to \infty} \|w^{q_j} - w^{r_j}\| = 0,$$

and from Corollary 5.13,

$$\lim_{j \to \infty} \|(Z^{q_j})^T \gamma^{q_j} - (Z^{r_j})^T \gamma^{r_j}\|_2 = 0.$$

Therefore, for any positive value ρ, there exists j large enough such that

$$\|(Z^{q_j})^T \gamma^{q_j} - (Z^{r_j})^T \gamma^{r_j}\|_2 \leq \rho.$$

Now, for large enough j,

$$
\begin{aligned}
\|(Z^{q_j})^T \gamma^{q_j}\|_2 &= \|(Z^{q_j})^T \gamma^{q_j} - (Z^{r_j})^T \gamma^{r_j} + (Z^{r_j})^T \gamma^{r_j}\|_2 \\
&\leq \|(Z^{q_j})^T \gamma^{q_j} - (Z^{r_j})^T \gamma^{r_j}\|_2 + \|(Z^{r_j})^T \gamma^{r_j}\|_2 \\
&\leq \rho + \Lambda.
\end{aligned}
$$

Since there are appropriately large (but finite) values of j for which the above condition holds for any positive value of ρ, it must necessarily be true that, for large enough j,

$$\|(Z^{q_j})^T \gamma^{q_j}\|_2 \leq \Lambda,$$

which contradicts the initial assumption in the proof. Therefore, that assumption was invalid, and the result follows. □

The next result finally establishes convergence of the algorithm to a stationary point.

Lemma 5.16 *Given Assumptions 5.1, 5.2, and 5.3, the iterates defined by Algorithm 4.1 satisfy*

$$\lim_{k\to\infty} \|(Z^k)^T\gamma^k\|_2 \leq \Lambda.$$

If $\mathcal{S}$ is finite, the result follows from Lemma 5.14. Otherwise, w^k, and therefore $\|(Z^k)^T\gamma^k\|_2$, will only change values on successful iterations, and

$$\lim_{k\to\infty} \|(Z^k)^T\gamma^k\|_2 = \lim_{k\in\mathcal{S}} \|(Z^k)^T\gamma^k\|_2,$$

and the result follows from Lemma 5.15. □

5.4 Convergence to a First Order Point

The goal of this section is to prove that the algorithm will eventually approach a first order point of the penalty function. This is established by examining how the algorithm handles type two points. Recall that a type two point appears to be close to a stationary, non-first order point.

The proof requires several steps.

1. If the algorithm is approaching a stationary, non-first order point, then the iterate will eventually be classified as a type two point.

2. Given a type two point, there is a lower bound on the decrease in the model function along a dropping direction, and hence on the decrease in the penalty function for a successful trust region dropping step.

3. Over a sequence of type two points, the trust region radius is bounded away from zero, so a successful iteration will eventually be performed.

4. The penalty function decrease on successful iterations is bounded away from zero.

5. Dropping steps cannot be performed infinitely often.

The results presented here are modeled after those in the previous section, and are inspired by Theorem 1 in [34].

5.4.1 Approaching a Type Two Point

In this section, an earlier result, Lemma 5.12, is used to establish that, if the algorithm is approaching a stationary, non-first order point, then the algorithm will eventually classify the iterate as a type two point.

In theory, a multiplier is out of kilter when it falls anywhere outside its optimal range. However, in practice, within the algorithm, a multiplier is considered out of kilter only when it is *safely* out of kilter. For the necessary optimality range $[-\mu, \mu]$, the multiplier estimate is considered in kilter if it is in the range $[-(1+\beta_\Psi)\mu, (1+\beta_\Psi)\mu]$, for an algorithm tolerance β_Ψ. Similarly, if the necessary optimality range is $[0, \mu]$ then the multiplier is considered in kilter if it lies in the range $[-\beta_\Psi, (1+\beta_\Psi)\mu]$. This generalization is used in an attempt to avoid dropping an activity whose multiplier is near one of the endpoints of the optimality ranges and its estimate is slightly out of kilter simply because the iterate is not yet close enough to the stationary, non-first order point.

Therefore, in order for a iterate to be classified as a type two point, the out of kilter multiplier must be safely out of kilter. This leads to the following definitions and assumptions before we restate the result. Note that the first assumption is also used in Theorem 1 in [34].

Assumption 5.4 *Assume that the number of stationary, non-first order points is finite.*

Definition 5.4 *Let $\hat{\Psi}$ refer to the multipliers at a stationary, non-first order point $\hat{w}$.*

1. *Let $\hat{\delta}_{\Psi}$ be the smallest magnitude by which a multiplier $\hat{\Psi}$ is out of kilter at a stationary, non-first order point.*

2. *Let δ_{mult}, a finite value, be the minimum of $\hat{\delta}_{\Psi}$ over all stationary, non-first order points.*

The following assumption on the tolerance β_{Ψ} is required as well.

Assumption 5.5 *Assume that the tolerance β_{Ψ} satisfies $\beta_{\Psi} < \delta_{mult}$.*

Lemma 5.17 *Given Assumptions 5.1 - 5.5, if there exists a subsequence $\{q_j\}$ for which $\|(Z^{q_j})^T\gamma^{q_j}\|_2 \le \Lambda$ and $\lim_{j\to\infty} w^{q_j} = \hat{w}$, for a stationary, non-first order point $\hat{w}$, then, w^k, for some k, will be eventually classified as a type two point.*

Proof: From Lemma 5.12, it follows that the set of ϵ-activities at w^{q_j} will eventually match the set of exact activities at $\hat{w}$. Therefore, it follows that

$$\mathcal{A}^{q_j}(\epsilon) \to \hat{\mathcal{A}} \text{ and } \gamma^{q_j} \to \hat{\gamma},$$

where $\hat{\mathcal{A}}$ and $\hat{\gamma}$ are evaluated at $\hat{w}$, and

$$\Psi^{q_j} \to \hat{\Psi}.$$

Since at least one of the multipliers at $\hat{w}$ is out of kilter, it must be true that, for j large enough, a component of Ψ^{q_j} must be out of kilter as well. Now, since $\Psi^{q_j} \to \hat{\Psi}$, after a finite number of iterations, some multiplier estimate will be safely out of kilter, due to Assumption 5.5. Therefore, this iterate will be classified as a type two point. □

5.4.2 A Bound on the Model Function Decrease

For convenience, superscript k indicating the iteration number is omitted because the results presented deal with a single iteration.

The model function decrease along dropping direction $d_{\mathcal{D}}$ is examined, where $d_{\mathcal{D}}$ is assumed to satisfy the following condition.

Assumption 5.6 *Dropping direction $d_{\mathcal{D}}$ is defined to drop a single activity. It satisfies $Ad_{\mathcal{D}} = -sign[\Psi_j]e_j$, where multiplier Ψ_j is safely out of kilter. In particular, assume that $d_{\mathcal{D}}$ satisfies*

$$d_{\mathcal{D}} = -sign[\Psi_j]Z_jZ_j^T\nabla a_j,$$

where ∇a_j is the gradient of the ϵ-activity being dropped, $\mathcal{A}_j$ is $\mathcal{A}$ with ∇a_j removed, and Z_j satisfies $Z_j = [Zz]$ and $\mathcal{A}_j^T Z_j = 0$.

The value of the model function along $d_{\mathcal{D}}$ is examined next. The ϵ-activity sets indicated below correspond to the current iterate, and have not been updated to reflect the change due to the dropped activity. This explains the need for the inclusion of the term $\Delta\varphi_{\mathcal{D}}(\alpha)$, which is defined below, on the last line,

$$\begin{aligned}
\varphi(\alpha d_{\mathcal{D}}) \quad = \quad & F(w) + \alpha d_{\mathcal{D}}^T\nabla F(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T\nabla^2 F(w)d_{\mathcal{D}} \\
& -\mu \sum_{i\in T'(\epsilon)} \min(G_i(w) + \alpha d_{\mathcal{D}}^T\nabla G_i(w) + \alpha^2 d_{\mathcal{D}}^T\nabla^2 G_i(w)d, 0) \\
& -\mu \sum_{i\in T^0(\epsilon)} \min(G_i(w), 0) \\
& +\mu \sum_{i\in P_\lambda'(\epsilon)\cup P_g'(\epsilon)} |\min(\lambda_i(w) + \alpha d_{\mathcal{D}}^T\nabla\lambda_i(w), \\
& \qquad\qquad g_i(w) + \alpha d_{\mathcal{D}}^T\nabla g_i(w) + \alpha^2 d_{\mathcal{D}}^T\nabla^2 g_i(w)d_{\mathcal{D}})| \\
& +\mu \sum_{i\in P_\lambda^0(\epsilon)} |\min(\lambda_i(w), g_i(w) + \alpha d_{\mathcal{D}}^T\nabla g_i(w) + \alpha^2 d_{\mathcal{D}}^T\nabla^2 g_i(w)d_{\mathcal{D}})| \\
& +\mu \sum_{i\in P_g^0(\epsilon)} |\min(\lambda_i(w) + \alpha d_{\mathcal{D}}^T\nabla\lambda_i(w), g_i(w))| \\
& +\mu \sum_{i\in P_\lambda^0(\epsilon)} |\min(\lambda_i(w), g_i(w) + \alpha d_{\mathcal{D}}^T\nabla g_i(w) + \alpha^2 d_{\mathcal{D}}^T\nabla^2 g_i(w)d_{\mathcal{D}})| \\
& +\mu \sum_{i\in P_=^\prime(\epsilon)} |\lambda_i(w) + d_{\mathcal{D}}^T\nabla\lambda_i(w) + \min(0, g_i(w) - \lambda_i(w))|
\end{aligned}$$

$$
\begin{aligned}
&+\mu \sum_{i\in P^0_=(\epsilon)} |\min(\lambda_i(w), g_i(w))| \\
&+\mu \sum_{i\in M'(\epsilon)} |c_i(w) + \alpha d_{\mathcal{D}}^T \nabla c_i(w) + \alpha^2 d_{\mathcal{D}}^T \nabla^2 c_i(w) d_{\mathcal{D}}| \\
&+\mu \sum_{i\in M^0} |c_i(w)| + \mu\Delta\varphi_{\mathcal{D}}(\alpha).
\end{aligned}
$$

The following assumption on the step size α is made so that the model function (and later the penalty function) can be examined more easily.

Assumption 5.7 *We shall consider only $\alpha \le \alpha_1$, where α_1 is the first breakpoint along $d_{\mathcal{D}}$ associated with the inactivities at w.*

Under this assumption, which was inherent in part 1 of the proof of Theorem 1 in [34], the model function can be written more compactly.

$$
\varphi(\alpha d_{\mathcal{D}}) = \varphi(0) + \alpha d_{\mathcal{D}}^T \gamma + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B(w) d_{\mathcal{D}} + \mu\Delta\varphi_{\mathcal{D}}(\alpha).
$$

The term $\Delta\varphi_{\mathcal{D}}$ is defined by removing the dropped activity from its active set and placing it in the appropriate inactivity set. Its specifics are listed below.

Definition 5.5 *The value of the function $\Delta\varphi_{\mathcal{D}}$ depends on the type of activity that was dropped in defining $d_{\mathcal{D}}$.*

- *If $j \in T^0(\epsilon)$, then*

$$
\begin{aligned}
\Delta\varphi_{\mathcal{D}}(\alpha) &= \min(G_j(w), 0) - \min(G_j(w) + \alpha d_{\mathcal{D}}^T \nabla G_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_j(w) d_{\mathcal{D}}, 0) \\
&= (\varsigma_j(0) - \varsigma_j(\alpha))G_j(w) - \alpha\varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla G_j(w) \\
&\quad -\frac{1}{2}\alpha^2 \varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla^2 G_j(w) d_{\mathcal{D}},
\end{aligned}
$$

where $\varsigma_j(\alpha) = neg[G_j(w) + \alpha d_{\mathcal{D}}^T \nabla G_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 G_j(w) d_{\mathcal{D}}]$.

- *If $j \in P^0_\lambda(\epsilon)$, then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -(|\lambda_j(w)| - |\lambda_j(w) + \alpha d_{\mathcal{D}}^T \nabla\lambda_j(w)|) \\ &= -(\lambda_j(w)(\varrho_j(0) - \varrho_j(\alpha)) - \alpha\varrho_j(\alpha) d_{\mathcal{D}}^T \nabla\lambda_j(w)),\end{aligned}$$

where $\varrho_j(\alpha) = sign[\lambda_j(w) + \alpha d_{\mathcal{D}}^T \nabla\lambda_j(w)]$.

- *If $j \in P^0_g(\epsilon)$, then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -(|g_j(w)| - |g_j(w) + \alpha d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}|) \\ &= -(g_j(w)(\varrho_j(0) - \varrho_j(\alpha)) - \alpha\varrho_j(\alpha) d_{\mathcal{D}}^T \nabla g_j(w) - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}),\end{aligned}$$

where $\varrho_j(\alpha) = sign[g_j(w) + \alpha d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}]$.

- *If $j \in P'_=(\epsilon)$, then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -\varrho_j(\min(0, g_j(w) - \lambda_j(w)) \\ &\qquad - \min(0, g_j(w) - \lambda_j(w) + \alpha d_{\mathcal{D}}^T(\nabla g_j(w) - \nabla\lambda_j(w)) \\ &\qquad\quad + \frac{1}{2}\alpha^2 \nabla^2 g_j(w) d_{\mathcal{D}})) \\ &= -\varrho_j((g_j(w) - \lambda_j(w))(\varsigma_j(0) - \varsigma_j(\alpha)) - \alpha\varsigma_j(\alpha) d_{\mathcal{D}}^T(\nabla g_j(w) - \nabla\lambda_j(w)) \\ &\quad -\frac{1}{2}\alpha^2 \varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}),\end{aligned}$$

where

$$\begin{aligned}\varsigma_j(\alpha) &= neg[g_j(w) - \lambda_j(w) + \alpha d_{\mathcal{D}}^T(\nabla g_j(w) - \nabla\lambda_j(w)) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}] \\ \varrho_j &= sign[\lambda_j(w)].\end{aligned}$$

- *If $j \in P^0_=(\epsilon)$ and $\lambda_j(w)$ is being dropped, then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -(|\min(\lambda_j(w), g_j(w))| - |\min(\lambda_j + \alpha d_{\mathcal{D}}^T \nabla\lambda_j(w), g_j(w))|) \\ &= -(\lambda_j(w)(\varrho_j(0) - \varrho_j(\alpha)) + (g_j(w) - \lambda_j(w))(\varrho_j(0)\varsigma_j(0) - \varrho_j(\alpha)\varsigma_j(\alpha)) \\ &\qquad - \alpha\varrho_j(\alpha)\varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla\lambda_j(w)),\end{aligned}$$

where

$$\begin{aligned}\varrho_j(\alpha) &= sign[\min(\lambda_j(w) + \alpha d_{\mathcal{D}}^T \nabla \lambda_j(w), g_j(w))] \\ \varsigma_j(\alpha) &= neg[g_j(w) - \lambda_j(w) - \alpha d_{\mathcal{D}}^T \nabla \lambda_j(w)].\end{aligned}$$

- *If* $j \in P^0_{=}(\epsilon)$ *and* $g_j(w)$ *is being dropped, then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -(|\min(\lambda_j(w), g_j(w))| \\ &\qquad - |\min(\lambda_i, g_j(w) + \alpha d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}})|) \\ &= -(\lambda_j(\varrho_j(0) - \varrho_j(\alpha)) + (g_j(w) - \lambda_j(w))(\varrho_j(0)\varsigma_j(0) - \varrho_j(\alpha)\varsigma_j(\alpha)) \\ &\qquad + \alpha\varrho_j(\alpha)\varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 \varrho_j(\alpha)\varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}),\end{aligned}$$

where

$$\begin{aligned}\varrho_j(\alpha) &= sign[\min(\lambda_j(w), g_j(w) + \alpha d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}})] \\ \varsigma_j(\alpha) &= neg[g_j(w) - \lambda_j(w) + \alpha d_{\mathcal{D}}^T \nabla g_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}]\end{aligned}$$

- *If* $j \in M^0(\epsilon)$, *then*

$$\begin{aligned}\Delta\varphi_{\mathcal{D}}(\alpha) &= -(|c_j(w)| - |c_j(w) + \alpha d_{\mathcal{D}}^T \nabla c_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_j(w) d_{\mathcal{D}}|) \\ &= -(c_j(w)(\varrho_j(0) - \varrho_j(\alpha)) - \alpha\varrho_j(\alpha) d_{\mathcal{D}}^T \nabla c_j(w) \\ &\qquad - \frac{1}{2}\alpha^2 \varrho_j(\alpha) d_{\mathcal{D}}^T \nabla^2 g_j(w) d_{\mathcal{D}}),\end{aligned}$$

where $\varrho_j(\alpha) = sign[c_j(w) + \alpha d_{\mathcal{D}}^T \nabla c_j(w) + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T \nabla^2 c_j(w) d_{\mathcal{D}}]$.

Each case is investigated individually. The results presented in Lemmas 5.18 - 5.22 are inspired by part 1 of the proof of Theorem 1 in [34].

Lemma 5.18 *Given Assumptions 5.1 - 5.7, if* $d_{\mathcal{D}} = -sign[\Psi_j^G] Z_j Z_j^T \nabla G_j(w)$ *for some* $j \in T^0(\epsilon)$, *then*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \alpha\delta_{\Psi_j^G} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where

$$\delta_{\Psi_j^G} = \begin{cases} |\Psi_j^G| & \text{if } \Psi_j^G < 0 \\ \Psi_j^G - \mu & \text{if } \Psi_j^G > \mu, \end{cases}$$

and

$$B_{\mathcal{D}}(\alpha) = B(w) - \mu\varsigma_j(\alpha)\nabla^2 G_j(w).$$

Proof: From above, we have that

$$\begin{aligned} \varphi(\alpha d_{\mathcal{D}}) - \varphi(0) &= \mu G_j(w)(\varsigma_j(0) - \varsigma_j(\alpha)) + \alpha(d_{\mathcal{D}}^T\gamma - \mu\varsigma_j(\alpha) d_{\mathcal{D}}^T \nabla G_j(w)) \\ &+ \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}. \end{aligned}$$

Note that, for some u in the reduced space,

$$\begin{aligned} d_{\mathcal{D}}^T\gamma &= d_{\mathcal{D}}^T(\mathcal{A}\Psi + Zu) \\ &= d_{\mathcal{D}}^T\mathcal{A}\Psi + 0 \\ &= -\text{sign}[\Psi_j^G]\Psi_j^G \\ &= -|\Psi_j^G| \end{aligned}$$

and $d_{\mathcal{D}}^T \nabla G_j(w) = -\text{sign}[\Psi_j^G]$. Therefore,

$$\begin{aligned} \varphi(\alpha d_{\mathcal{D}}) - \varphi(0) &= \mu G_j(w)(\varsigma_j(0) - \varsigma_j(\alpha)) - \alpha(|\Psi_j^G| - \mu\varsigma_j(\alpha)\text{sign}[\Psi_j^G]) \\ &+ \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}. \end{aligned}$$

Examining the first terms on the right hand side yields

$$\begin{aligned} G_j(w)(\varsigma_j(0) - \varsigma_j(\alpha)) &= \begin{cases} 0 & \text{if } \varsigma_j(0) = \varsigma_j(\alpha) \\ G_j(w) & \text{if } \varsigma_j(0) = 1 \\ -G_j(w) & \text{if } \varsigma_j(0) = 0 \end{cases} \\ &\leq 0, \end{aligned}$$

and

$$|\Psi_j^G| - \mu\varsigma_j(\alpha)\text{sign}[\Psi_j^G] = \begin{cases} |\Psi_j^G| - \mu\varsigma_j(\alpha) \geq \Psi_j - \mu & \text{if } \Psi_j^G > \mu \\ |\Psi_j^G| + \mu\varsigma_j(\alpha) \geq |\Psi_j| & \text{if } \Psi_j^G < 0. \end{cases}$$
$$\geq \delta_{\Psi_j^G}.$$

Therefore,

$$\varphi(\alpha d_{\mathcal{D}}) - \varphi(0) \leq -\alpha\delta_{\Psi_j^G} + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}.$$

The result follows immediately. □

Lemma 5.19 *Given Assumptions 5.1 - 5.7, if* $d_{\mathcal{D}} = -sign[\Psi_j^g]Z_jZ_j^T\nabla g_j(w)$ *for some* $j \in P_g^0(\epsilon)$*, then,*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \alpha\delta_{\Psi_j^g} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where

$$\delta_{\Psi_j^G} = |\Psi_j^G| - \mu$$

and

$$B_{\mathcal{D}}(\alpha) = B(w) - \mu\varrho_j(\alpha)\nabla^2 g_j(w).$$

Proof: From above, we have that

$$\varphi(\alpha d_{\mathcal{D}}) - \varphi(0) = -\mu g_j(w)(\varrho_j(0) - \varrho_j(\alpha)) + \alpha(d_{\mathcal{D}}^T\gamma - \mu\varrho_j(\alpha)d_{\mathcal{D}}^T\nabla g_j(w))$$
$$+\frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}.$$

Note that, for some u in the reduced space,

$$d_{\mathcal{D}}^T\gamma = d_{\mathcal{D}}^T(\mathcal{A}\Psi + Zu) = d_{\mathcal{D}}^T\mathcal{A}\Psi = -\text{sign}[\Psi_j^g]\Psi_j^g = -|\Psi_j^g|.$$

and $d_{\mathcal{D}}^T \nabla g_j(w) = -\text{sign}[\Psi_j^g]$. Therefore,

$$\begin{aligned}\varphi(\alpha d_{\mathcal{D}}) - \varphi(0) &= -\mu g_j(w)(\varrho_j(0) - \varrho_j(\alpha)) - \alpha(|\Psi_j^G| - \mu\varrho_j(\alpha)\text{sign}[\Psi_j^g]) \\ &\quad + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}.\end{aligned}$$

Examining the terms on the right hand side yields

$$\begin{aligned} g_j(w)(\varrho_j(0) - \varrho_j(\alpha)) &= \begin{cases} 0 & \text{if } \rho_j(0) = \varrho_j(\alpha) \\ 2G_j(w) & \text{if } \varrho_j(0) = 1 \\ -2G_j(w) & \text{if } \varrho_j(0) = -1 \end{cases} \\ &\geq 0, \end{aligned}$$

and

$$|\Psi_j^g| - \mu\varrho_j(\alpha)\text{sign}[\Psi_j^g] \geq |\Psi_j^g| - \mu = \delta_{\Psi_j^G}.$$

Therefore,

$$\varphi(\alpha d_{\mathcal{D}}) - \varphi(0) \leq -\alpha\delta_{\Psi_j^g} + \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}. \square$$

Lemma 5.20 *Given Assumptions 5.1 - 5.7, if* $d_{\mathcal{D}} = -sign[\Psi_j^\lambda] Z_j Z_j^T \nabla\lambda_j(w)$ *for some* $j \in P_\lambda^0(\epsilon)$, *then,*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \alpha\delta_{\Psi_j^\lambda} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where

$$\delta_{\Psi_j^\lambda} = |\Psi_j^\lambda| - \mu$$

and $B_{\mathcal{D}}(\alpha) = B(w)$.

Proof: The result is proven using the same technique as for $j \in P_g^0(\epsilon)$. $\square$

Recall that $P_=^\prime(\epsilon) = P_=^+(\epsilon) \cup P_=^-(\epsilon)$ where the $\pm$ indicates the common sign of $\lambda(w)$ and $g_j(w)$. For our purposes here, the sets must be separated, just as they were in determining optimality conditions.

Lemma 5.21 *Given Assumptions 5.1 - 5.7, if* $d_{\mathcal{D}} = -sign[\Psi_j^{g-\lambda}]Z_jZ_j^T(\nabla g_j(w) - \nabla\lambda_j(w))$ *for some* $j \in P_{=}^{-}(\epsilon)$*, then,*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \alpha\delta_{\Psi_j^{g-\lambda}} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where

$$\delta_{\Psi_j^{g-\lambda}} = \begin{cases} |\Psi_j^{g-\lambda}| & \text{if } \Psi_j^{g-\lambda} < 0 \\ \Psi_j^{g-\lambda} - \mu & \text{if } \Psi_j^{g-\lambda} > \mu, \end{cases}$$

and

$$B_{\mathcal{D}}(\alpha) = B(w) - \mu\varsigma_j(\alpha)\nabla^2 g_j(w).$$

Proof: The result is proven using the same technique as for $j \in T^0(\epsilon)$. □

Lemma 5.22 *Given Assumptions 5.1 - 5.7, if* $d_{\mathcal{D}} = -sign[\Psi_j^c]Z_jZ_j^T\nabla c_j(w)$ *for some* $j \in M^0(\epsilon)$*, then,*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \alpha\delta_{\Psi_j^c} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where

$$\delta_{\Psi_j^c} = |\Psi_j^c| - \mu$$

and

$$B_{\mathcal{D}}(\alpha) = B(w) - \mu\varrho_j(\alpha)\nabla^2 c_j(w).$$

Proof: The result is proven using the same technique as for $j \in P_g^0$. □

Further assumptions specific to the bilevel penalty function are required for the analysis of the convergence of the algorithm. These assumptions do not apply to the implemented algorithm.

Assumption 5.8 *At any type two point* w*, assume that*

- *the set* $P^{+}_{\equiv}(\epsilon) = \emptyset$, *and*
- *the set* $P^{0}_{\equiv}(\epsilon) = \emptyset$.

The first condition is a required condition at a nondegenerate minimum point of the penalty function. Essentially, we are assuming that this type of constraint is never violated at a stationary, non-first order point. Recall that if the set is not empty, there is a range of values for which a dropping direction can satisfy one of two conditions for descent. Requiring $P^{+}_{\equiv}(\epsilon) = \emptyset$ removes an ambiguity from the proving process.

The second condition is equivalent to requiring strict complementarity in the lower level problem of the original bilevel problem. This condition is required for many algorithms for bilevel problems.

Therefore, we can summarize our results as follows.

Corollary 5.23 *Given Assumptions 5.1 - 5.8, if* w *is a type two point, then dropping direction* $d_{\mathcal{D}}$ *must satisfy*

$$\varphi(\alpha d_{\mathcal{D}}) - \varphi(0) \geq \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}},$$

where δ_{Ψ_j} *is the magnitude by which the multiplier corresponding to* $d_{\mathcal{D}}$ *is out of kilter.*

Proof: Follows immediately from the above results. □

For convenience, we introduce the function $\theta(\alpha)$.

Definition 5.6 *For* $0 \leq \alpha \leq \alpha_1$,

$$\theta(\alpha) = \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}} d_{\mathcal{D}}.$$

Additional terminology is now required for second order information. Note the similarities between the following set of definitions and those in Definition 5.1.

Definition 5.7 *For any iteration k, the following terminology is introduced for second order change in θ.*

1. *Let*

$$\Omega^k(\theta, \alpha d_{\mathcal{D}}) = \frac{d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}}}{\|d_{\mathcal{D}}\|_2^2}.$$

2. *Define the maximum over α as*

$$\Omega_{\mathcal{D}}^k = \max_{0 \leq \alpha \leq \alpha_1^k} \Omega^k(\theta, \alpha d_{\mathcal{D}}).$$

3. *Define a maximum term over all previous iterations as*

$$\tau_{\mathcal{D}}^k = 1 + \max_{1 \ldots k} \Omega_{\mathcal{D}}^k.$$

The next assumption on the curvature of the model function along a dropping direction is analogous to Assumption 5.2 for curvature along the generalized Cauchy direction.

Assumption 5.9 *The following condition on the curvature measurements, over all iterations, is assumed to be true:*

$$\sum_{k=1}^{\infty} \frac{1}{\tau_{\mathcal{D}}^k} = +\infty.$$

A consequence of this assumption is that $\tau_{\mathcal{D}}^k$ is bounded above, i.e. there exists $\tau_{\mathcal{D}}^{\max} > 0$ such that $\tau_{\mathcal{D}}^k \leq \tau_{\mathcal{D}}^{\max}$ for all k, in addition to being bounded below by one (by definition).

Using these definitions,

$$\begin{aligned}
\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) &\geq \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 d_{\mathcal{D}}^T B_{\mathcal{D}}(\alpha) d_{\mathcal{D}} \\
&= \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 \|d_{\mathcal{D}}\|_2^2 \Omega(\theta, \alpha d_{\mathcal{D}}) \\
&\geq \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 \|d_{\mathcal{D}}\|_2^2 \Omega_{\mathcal{D}} \\
&\geq \alpha \delta_{\Psi_j} - \frac{1}{2}\alpha^2 \|d_{\mathcal{D}}\|_2^2 \tau_{\mathcal{D}}.
\end{aligned}$$

Next, look at $d_{\mathcal{D}}$ more closely. Let ∇a_j denote the gradient of the ϵ-activity that is dropped along $d_{\mathcal{D}}$. Note that $Z_j^T Z_j = I$.

$$\begin{aligned}
||d_{\mathcal{D}}||_2^2 &= d_{\mathcal{D}}^T d_{\mathcal{D}} \\
&= (-\text{sign}[\Psi_j] Z_j Z_j^T \nabla a_j)^T (-\text{sign}[\Psi_j] Z_j Z_j^T \nabla a_j) \\
&= (\pm 1)^2 \nabla a_j^T Z_j Z_j^T Z_j Z_j^T \nabla a_j \\
&= \nabla a_j^T Z_j Z_j^T \nabla a_j \\
&= ||Z_j^T \nabla a_j||_2^2.
\end{aligned}$$

Recall that $Z_j = [Zz]$, where $Z^T \nabla a_j = 0$ and $z^T z = 1$. Therefore,

$$\begin{aligned}
||d_{\mathcal{D}}||_2^2 &= \left\| \begin{bmatrix} Z^T \\ z^T \end{bmatrix} \nabla a_j \right\|_2^2 \\
&= \left\| \begin{bmatrix} Z^T \nabla a_j \\ z^T \nabla a_j \end{bmatrix} \right\|_2^2 \\
&= \left\| \begin{bmatrix} 0 \\ z^T \nabla a_j \end{bmatrix} \right\|_2^2 \\
&= (z^T \nabla a_j)^2 \\
&\leq ||z||_2^2 \, ||\nabla a_j||_2^2 \\
&= ||\nabla a_j||_2^2.
\end{aligned}$$

Because the individual functions comprising the penalty functions are assumed to be well-behaved and the penalty function is assumed to be bounded over the interval in Assumption 5.2, it is reasonable to further assume the following:

Assumption 5.10 *The norms of the gradients of the individual functions are bounded above.*

Definition 5.8 *Let A_m be large enough so that $\|\nabla a_j\|_2 \leq A_m$ is true for all activities over all iterations.*

Lemma 5.24 *Given Assumptions 5.1 - 5.10, for a type two point w, it follows that*

$$\varphi(0) - \varphi(\alpha d_{\mathcal{D}}) \geq \vartheta_{\mathcal{D}}(\alpha),$$

where

$$\vartheta_{\mathcal{D}}(\alpha) = \alpha\delta_{\Psi_j} - \frac{1}{2}\alpha^2 A_{\max}^2 \tau_{\mathcal{D}}.$$

Proof: Follows immediately from the definitions of θ, A_m and $\tau_{\mathcal{D}}$. □

Next, to determine the step to take along the dropping direction, consider solving the following univariate problem, a simplified version of the trust region subproblem

$$\text{TH1:} \quad \min \vartheta_{\mathcal{D}}(\alpha) \text{ s.t. } 0 \leq \alpha \leq \alpha_1 \text{ and } \alpha \leq \Delta/\|d_{\mathcal{D}}\|_\infty.$$

Very small values of α provide a decrease in ϑ, so the minimum cannot occur at $\alpha = 0$. Therefore, the minimum will occur at α_ϑ, corresponding to one of the following situations:

1. at the unconstrained minimum of ϑ, i.e. at
$$\alpha_{\min} = \frac{\delta_{\Psi_j}}{A_m^2 \tau_{\mathcal{D}}},$$
2. at the upper bound α_1, or
3. at the trust region upper bound $\alpha_\Delta = \Delta/\|d_{\mathcal{D}}\|_\infty$.

The following results echo results presented in the first stage of the convergence proof (in Lemmas 5.1, 5.2 and 5.3).

Lemma 5.25 *Given Assumptions 5.1-5.10, if the solution of TH1 occurs at $\alpha_\vartheta = \alpha_{\min}$, then*

$$\vartheta_{\mathcal{D}}(\alpha_\vartheta) \geq \frac{1}{2}\frac{\delta_{\Psi_j}}{A_m^2 \tau_{\mathcal{D}}}.$$

Proof: Follows directly from substituting α_ϑ into ϑ. □

Lemma 5.26 *Given Assumptions 5.1-5.10, if the solution of TH1 occurs at $\alpha_\vartheta = \alpha_1$, then*

$$\vartheta_{\mathcal{D}}(\alpha_\vartheta) \geq \frac{1}{2}\alpha_1 \delta_{\Psi_j}.$$

Proof: Substituting α_ϑ into ϑ gives

$$\vartheta(\alpha_\vartheta) = \alpha_1 \delta_{\Psi_j} - \frac{1}{2}\alpha_1^2 A_m^2 \tau_{\mathcal{D}}.$$

Since $\alpha_{\min} \notin [0, \alpha_1]$, it follows that $\alpha_{\min} > \alpha_1$,

$$\alpha_1 < \frac{\delta_{\Psi_j}}{A_m^2 \tau_{\mathcal{D}}} \Rightarrow \delta_{\Psi_j} > \alpha_1 A_m^2 \tau_{\mathcal{D}}.$$

Therefore,

$$\vartheta_{\mathcal{D}} \geq \alpha_1 \delta_{\Psi_j} - \frac{1}{2}\alpha_1 \delta_{\Psi_j} = \frac{1}{2}\alpha_1 \delta_{\Psi_j}. \qquad \square$$

Lemma 5.27 *Given Assumptions 5.1-5.10, if the solution of TH1 occurs at $\alpha_\vartheta = \alpha_\Delta$, then*

$$\vartheta_{\mathcal{D}}(\alpha_\vartheta) \geq \frac{\Delta \delta_{\Psi_j}}{2A_m}.$$

Proof: Follows immediately from substituting α_ϑ into ϑ and noting that $\alpha_{\min} > \alpha_\Delta$. □

Combining the three previous results gives the following.

Corollary 5.28 *Given Assumptions 5.1-5.10,*

$$\vartheta_{\mathcal{D}}(\alpha_\vartheta) \geq \frac{1}{2}\delta_{\Psi_j} \min\left(\frac{\delta_{\Psi_j}}{A_m^2 \tau_{\mathcal{D}}}, \alpha_1, \frac{\Delta}{A_m}\right).$$

Proof: Follows immediately by noting that one of the conditions listed Lemma 5.25, 5.26 and 5.27 must be satisfied. □

Corollary 5.29 *Given Assumptions 5.1-5.10,*

$$\varphi(0) - \varphi(\alpha_\vartheta d_\mathcal{D}) \geq \frac{1}{2}\delta_{\Psi_j} \min\left(\frac{\delta_{\Psi_j}}{A_m^2 \tau_\mathcal{D}}, \alpha_1, \frac{\Delta}{A_m}\right).$$

Proof: Follows immediately from Corollary 5.28 and Lemma 5.24. □

Now, let $\alpha_\mathcal{D}$ be the step along $d_\mathcal{D}$ taken to the minimum of $\varphi(\alpha d_\mathcal{D})$.

Corollary 5.30 *Given Assumptions 5.1-5.10,*

$$\varphi(0) - \varphi(\alpha_\mathcal{D}) \geq \frac{1}{2}\delta_{\Psi_j} \min\left(\frac{\delta_{\Psi_j}}{A_m^2 \tau_\mathcal{D}}, \alpha_1, \frac{\Delta}{A_m}\right).$$

Proof: From the definition of $\alpha_\mathcal{D}$, $\varphi(\alpha_D d_\mathcal{D}) \leq \varphi(\alpha_\vartheta)$. The result follows immediately from Corollary 5.29. □

All the above results have been presented for the current iteration, so the superscript k indicating the iteration number has been omitted. For clarity, I will now restate the above result with the required superscript,

$$\varphi^k(0) - \varphi^k(\alpha_\mathcal{D}^k d_\mathcal{D}) \geq \frac{1}{2}\delta_{\Psi_j}^k \min\left(\frac{\delta_{\Psi_j}^k}{A_m^2 \tau_\mathcal{D}^k}, \alpha_1^k, \frac{\Delta^k}{A_m}\right).$$

Corollary 5.31 *Given Assumptions 5.1 - 5.10, if k is a successful trust region iteration, then*

$$p_\mu(w^k) - p_\mu(w^k + \alpha_\mathcal{D}^k d_\mathcal{D}^k) \geq \frac{1}{2} b_1 \delta_{\Psi_j}^k \min\left(\frac{\delta_{\Psi_j}^k}{A_m^2 \tau_\mathcal{D}^k}, \alpha_1^k, \frac{\Delta^k}{A_m}\right),$$

where b_1 is the algorithm parameter used to determine a successful trust region iteration.

5.4.3 A Bound on the Trust Region Radius

This section will proceed very much like Section 5.3.2 in the first stage of the convergence results. In fact, some of the results from that section apply directly to the current situation.

Lemma 5.32 *Given Assumptions 5.1 - 5.10, there exists $\delta_\alpha > 0$ such that for all iterations from type two points,*

$$\alpha_1^k \;\geq\; \delta_\alpha.$$

Proof: The proof of Lemma 5.7 applies here as well. □

Lemma 5.33 *Given Assumptions 5.1 - 5.10, for all type two points w^k, there exists a constant $q_2 > 0$ such that*

$$|p_\mu(w^k + \alpha_{\mathcal{D}}^k d_{\mathcal{D}}^k) - \varphi^k(\alpha_{\mathcal{D}}^k d_{\mathcal{D}})| \leq q_2(\Delta^k)^2 \tau_{\mathcal{D}}^k.$$

Proof: The reader is referred to the proof of Lemma 5.8. With the appropriate corrections to the ϵ-active and ϵ-inactive sets to reflect the change in the dropped activity, the same proof is applicable to the current situation. □

The next result is analogous to Lemma 5.9 for the case of type one points. An additional assumption on β_Ψ, similar to Assumption 5.3, is required first.

Assumption 5.11 *The algorithm parameter β_Ψ is small enough that the following condition is satisfied:*

$$\beta_\Psi < \min\left(\delta_\alpha, 2A_m\tau^0\Delta^0, \frac{4A_m q_2 \tau^0 \Delta^0}{1-b_2}, 2\Delta^0\tau^0\sqrt{\frac{2q_2}{1-b_2}}\right).$$

Lemma 5.34 *Given Assumptions 5.1 - 5.11, and a sequence of type two points w^k, there exists a constant $\delta_{\mathcal{D}} > 0$ such that,*

$$\Delta^k \tau_{\mathcal{D}}^k \geq \delta_{\mathcal{D}}.$$

Proof: This proof uses the same technique used for Lemma 5.9. Define $\delta_{\mathcal{D}}$ as

$$\delta_{\mathcal{D}} = \frac{1}{2}\beta_{\Psi} \min\left(\frac{1}{A_m}, \frac{1-b_2}{2A_m q_2}, \sqrt{\frac{1-b_2}{2q_2}}\right). \tag{5.6}$$

Assume that the lemma is not true, i.e. that there exists an index J such that

$$\Delta^J \tau^J < \delta_{\mathcal{D}}. \tag{5.7}$$

The definition of $\delta_{\mathcal{D}}$ and Assumption 5.11 ensure that $\Delta^0\tau^0 \geq \delta_{\mathcal{D}}$, so $J \geq 1$. Without loss of generality, assume that J is the smallest index satisfying (5.7), i.e. $\Delta^k\tau^k \geq \delta_{\mathcal{D}}$ for $k = 0 : J-1$.

Define ξ^{J-1} as follows:

$$\begin{aligned}
\xi^{J-1} &= \left|\frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1\right| \\
&= \left|\frac{p_\mu(w^{J-1} + d_{\mathcal{D}}^{J-1}) - p_\mu(w^{J-1})}{\varphi^{J-1}(d_{\mathcal{D}}^{J-1}) - \varphi^{J-1}(0)} - 1\right| \\
&= \left|\frac{p_\mu(w^{J-1} + d_{\mathcal{D}}^{J-1}) - \varphi^{J-1}(d_{\mathcal{D}}^{J-1})}{\varphi^{J-1}(d_{\mathcal{D}}^{J-1}) - \varphi^{J-1}(0)}\right|,
\end{aligned}$$

since $p_\mu(w^{J-1}) = \varphi^{J-1}(0)$. Using the presented results and assumptions, it follows that

$$\xi^{J-1} \leq \frac{2q_2(\Delta^{J-1})^2\tau_{\mathcal{D}}^{J-1}}{\beta_{\Psi}\min\left(\frac{\beta_{\Psi}}{A_m^2\tau_{\mathcal{D}}^{J-1}}, \frac{\Delta^{J-1}}{A_m}, \alpha_1^{J-1}\right)} \leq \frac{2q_2(\Delta^{J-1})^2\tau_{\mathcal{D}}^{J-1}}{\beta_{\Psi}\min\left(\frac{\beta_{\Psi}}{A_m^2\tau_{\mathcal{D}}^{J-1}}, \frac{\Delta^{J-1}}{A_m}, \delta_\alpha\right)}.$$

Consider the following:

$$\begin{aligned}
\tau^{J-1}\Delta^{J-1} &\leq \tau^J\Delta^{J-1} && \text{since } \tau^{J-1} \leq \tau^J \\
&\leq 2\tau^J\Delta^J && \text{since } \Delta^{J-1} \leq 2\Delta^J \text{ even if iteration is unsuccessful} \\
&< 2\delta_{\mathcal{D}} && \text{from (5.7)} \\
&\leq \beta_{\Psi}/A_m && \text{from (5.6).}
\end{aligned}$$

Since $\beta_{\Psi} \geq \tau^{J-1}\Delta^{J-1}A_m$ and $\tau^{J-1}\Delta^{J-1} < 2\delta_{\mathcal{D}}$,

$$\xi^{J-1} \leq \frac{2q_2(\Delta^{J-1})^2\tau^{J-1}}{\beta_{\Psi}\min\left(\frac{\Delta^{J-1}}{A_m}, \delta_\alpha\right)} \leq \frac{4q_2\Delta^{J-1}\delta_{\mathcal{D}}}{\beta_{\Psi}\min\left(\frac{\Delta^{J-1}}{A_m}, \delta_\alpha\right)}.$$

Consider the two possibilities for the min term separately.

- $\Delta^{J-1}/A_m \leq \delta_\alpha$:

$$\xi^{J-1} \leq \frac{4A_m q_2\delta_{\mathcal{D}}}{\beta_{\Psi}} \leq \frac{4A_m q_2}{\beta_{\Psi}}\frac{\beta_{\Psi}(1-b_2)}{4A_m q_2} = 1 - b_2.$$

- $\delta_\alpha < \Delta^{J-1}/A_m$:

$$\xi^{J-1} \leq \frac{4q_2\Delta^{J-1}\delta_{\mathcal{D}}}{\beta_{\Psi}\delta_\alpha} \leq \frac{4q_2\Delta^{J-1}\tau_{\mathcal{D}}^{J-1}\delta_{\mathcal{D}}}{\beta_{\Psi}\delta_\alpha} \leq \frac{8q_2\delta_{\mathcal{D}}^2}{\beta_{\Psi}^2} \leq \frac{8q_2}{\beta_{\Psi}^2}\frac{(1-b_2)\beta_{\Psi}^2}{8q_2} = 1 - b_2.$$

Therefore, in either case, $\xi^{J-1} \leq 1 - b_2$, or

$$\left|\frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1\right| \leq 1 - b_2 \Rightarrow \frac{\chi_a^{J-1}}{\chi_p^{J-1}} - 1 \geq b_2 - 1 \Rightarrow \frac{\chi_a^{J-1}}{\chi_p^{J-1}} \geq b_2,$$

i.e. iteration $J-1$ is very successful and $\Delta^{J-1} \leq \Delta^J$. So,

$$\Delta^{J-1}\tau^{J-1} \leq \Delta^J\tau^{J-1} \leq \Delta^J\tau^J < \delta_{\mathcal{D}},$$

which contradicts the fact that J is the smallest index satisfying (5.7). The result follows immediately from the contradiction. □

5.4.4 A Bound on the Penalty Function Decrease

We can now prove the following result.

Lemma 5.35 *Given Assumptions 5.1 - 5.11, for any successful iteration from a type two point, there exists $\delta_{\mathcal{D}}^p > 0$ such that*

$$p_\mu(w^k) - p_\mu(w^k + \alpha_{\mathcal{D}}^k d_{\mathcal{D}}) \quad \geq \quad \delta_{\mathcal{D}}^p.$$

Proof: In Corollary 5.31, it was established that

$$p_\mu(w^k) - p_\mu(w^k + \alpha_{\mathcal{D}}^k d_{\mathcal{D}}^k) \geq \frac{1}{2} b_1 \delta_{\Psi_j}^k \min\left(\frac{\delta_{\Psi_j}^k}{A_m^2 \tau_{\mathcal{D}}^k}, \alpha_1^k, \frac{\Delta^k}{A_m}\right).$$

Now, from the definition of β_Ψ, Assumption 5.9, and Lemmas 5.32 and 5.34, it follows that

$$p_\mu(w^k) - p_\mu(w^k + \alpha_{\mathcal{D}}^k d_{\mathcal{D}}^k) \geq \frac{1}{2} b_1 \beta_\Psi \min\left(\frac{\beta_\Psi}{A_m^2 \tau_{\mathcal{D}}^{\max}}, \delta\alpha, \frac{\delta_{\mathcal{D}}}{A_m}\right) = \delta_{\mathcal{D}}^p. \square$$

5.4.5 Approaching a First Order Point

This section lays the groundwork for the final stage of the convergence proof, which is presented in the subsequent section.

Lemma 5.36 *Given Assumptions 5.1-5.11, only a finite number of dropping steps will be performed.*

Proof: Since p_μ is assumed to be bounded below, and the decrease in the penalty function is bounded away from zero on a successful trust region step, there can only be a finite number of successful dropping steps.

Since the trust region radius is reduced for unsuccessful trust region steps and it has been established that Δ is bounded away from zero, it follows that there can only be a finite number of unsuccessful dropping steps. $\square$

Lemma 5.37 *Given Assumptions 5.1 - 5.11, the algorithm will eventually approach a first order point.*

Proof: There are only a finite number of stationary, non-first order points (by Assumption 5.4) and there can only be a finite number of dropping steps. Therefore, after some point, the algorithm will no longer approach a stationary, non-first order point.

By Lemma 5.16, the iterates will approach a different stationary point. Eventually, the stationary point will be a first order point. □

Lemma 5.38 *Given Assumptions 5.1 - 5.11 and constant values of the algorithm parameters ϵ and Λ, all iterates will eventually be type three or type four points.*

Proof: From previous results, the algorithm will eventually approach a first order point. Using the same technique as in the proof of Lemma 5.17, it follows that the multiplier estimates at the iterates will eventually approach the multipliers at the first order point. Initially, an out of kilter estimate may be obtained, but eventually the estimate will lie in the range to be considered in kilter, due to the use of the bound β_{Φ}. □

5.5 Convergence to a Second Order Point

To establish that the algorithm will eventually converge to a second order point of the penalty function, the following results must be proven.

1. The algorithm will eventually attempt a Newton step.
2. A Newton step will eventually be successful.
3. The algorithm tolerances ϵ and Λ, which are reduced when an iterates has been misclassified as close to a stationary point, will only be reduced a finite number of times.
4. Eventually, all iterations are Newton steps.

5.5.1 Approaching a Type Four Point

From Section 4.2.5, recall that the Newton step satisfies $d_N = d_h + d_v$, where

$$\begin{aligned} d_h &= -Z(Z^T HZ)^{-1} Z^T \gamma \\ \text{and } d_v &= -A(A^T A)^{-1} \Phi(w + d_h). \end{aligned}$$

It has already been established that, for constant values of ϵ and Λ, the algorithm will eventually approach a first order point. This point may be a second order point (a possible solution of the penalty function) or a saddle point.

As in Coleman and Conn [34], the following definition and assumption are required.

Definition 5.9 *A second order point w is a strict second order point if none of the multipliers lie at their optimal boundary and if the reduced Hessian is positive definite.*

Assumption 5.12 *Assume that all first order points of the penalty function are strict second order points.*

Note that this assumption is made for the purposes of theoretical analysis of the algorithm, and is not applied to the implemented algorithm.

Corollary 5.39 *Given Assumptions 5.1 - 5.12, when approaching a strict second order point, the algorithm will eventually attempt a full Newton step.*

Proof: From Assumption 5.12, when the current iterate is close to a first order point of p_μ, it is actually close to a strict second order point, and the reduced Hessian is positive definite. Therefore, type three points are assumed to be never encountered. Combining this observation with Lemma 5.38, the algorithm will eventually classify an iterate as a type four point, and the Newton step will be attempted. □

Due to the positive definiteness of the reduced Hessian near a strict second order point, an assumption on the size of the inverse of the reduced Hessian, as in Lemma 1 in [34], can be made.

Assumption 5.13 *Assume that there exist strictly positive constants b_1^H and b_2^H such that for any direction h,*

$$b_1^H \|h\|_2^2 \leq h^T (Z^T H Z)^{-1} h \leq b_2^H \|h\|_2^2.$$

And, in particular, assume that $\|(Z^T H Z)^{-1}\|_2 \leq \delta_{HZ}$ for some constant $\delta_{HZ} > 0$.

At a type four point, the algorithm calculates the Newton step $d_N = d_h + d_v$. If it satisfies $\|d_n\|_\infty \leq \Delta$ and provide sufficient decrease in p_μ, then it is accepted as the trust region direction and the new iterate is calculated. Otherwise, the tolerances ϵ and Λ are reduced and the current iterate is reclassified because it is not as close to a second order point of p_μ as originally thought.

5.5.2 Success of the Newton Step

In this section, the value of $p_\mu(w + d_h + d_v)$ is analyzed to determine if a full Newton step will be successful when close to the strict second order point $\hat{w}$. This derivation is patterned after the corresponding step in Coleman and Conn [34].

For simplicity, the following definition is formalized.

Definition 5.10 *The point w is "close enough" to a minimum point $\hat{w}$ when the following conditions are satisfied.*

1. *Assumption 5.13 is satisfied.*

2. *The set of ϵ-activities at w is composed of the set of exact activities at $\hat{w}$.*

3. *The step $d_h + d_v$ is small enough so that no breakpoints are passed from w to $w + d_h + d_v$.*

Lemma 5.40 *Given Assumptions 5.1 - 5.13, when a type four point w is "close enough" to $\hat{w}$, there exists a positive constant L such that*

$$p_\mu(w + d_h + d_v) - p_\mu(w) \quad \leq \quad -L(\|Z^T\gamma\|_2^2 + \|\Phi\|_1).$$

Proof: This proof uses the technique of and terminology from the proof of Lemma 1 in [34]. From Definition 5.10, we can write the following. Note that γ and B are the gradient and Hessian, respectively, of the differentiable part of $p_\mu(w)$.

$$\begin{aligned}
p_\mu(w + d_h + d_v) \quad = \quad & p_\mu(w) + (d_h + d_v)^T\gamma + \frac{1}{2}(d_h + d_v)^T B(d_h + d_v) + o(\|d_h + d_v\|^2) \\
& -\mu \sum_{T^0(\epsilon)} (\min(0, G_i(w + d_h + d_v)) - \min(0, G_i(w))) \\
& +\mu \sum_{P_\lambda^0(\epsilon)} (|\lambda_i(w + d_h + d_v)| - |\lambda_i(w)|) \\
& +\mu \sum_{P_g^0(\epsilon)} (|g_i(w + d_h + d_v)| - |g_i(w)|) \\
& -\mu \sum_{P_=^-(\epsilon)} (\min(0, g_i(w + d_h + d_v) - \lambda_i(w + d_h + d_v)) \\
& \qquad - \min(0, g_i(w) - \lambda_i(w))) \\
& +\mu \sum_{M^0(\epsilon)} (|c_i(w + d_h + d_v)| - |c_i(w)|).
\end{aligned}$$

We will need to consider the component parts of $p_\mu(w + d_h + d_v)$ in more detail.

Consider the following:

$$\begin{aligned}
d_h^T\gamma &= -\gamma_Z^T H_Z^{-1}\gamma_Z \\
d_v^T\gamma &= d_v^T(A\Psi + Zu) \\
&= d_v^T A\Psi + d_v^T Zu
\end{aligned}$$

$$
\begin{aligned}
&= d_v^T A\Psi + \Phi(w+d_h)^T(A^TA)^{-1}A^TZu \\
&= d_v^T A\Psi + 0 \\
&= -\Phi(w+d_h)^T(A^TA)^{-1}A^TA\Psi \\
&= -\Phi(w+d_h)^T\Psi \\
&= -\sum \Psi_i\Phi_i(w+d_h) \\
&= -\sum \Psi_i(a_i(w) + d_h^T\nabla a_i(w) + \frac{1}{2}d_h^T\nabla^2 a_i d_h) + o(\|d_h\|^2) \\
&= -\sum \Psi_i(a_i(w) + \frac{1}{2}d_h^T\nabla^2 a_i d_h) + o(\|d_h\|^2) \\
&= -\sum_{T^0(\epsilon)} \Psi_i^G(G_i(w) + \frac{1}{2}d_h^T\nabla^2 G_i(w)d_h) - \sum_{P_\lambda^0(\epsilon)} \Psi_i^\lambda \lambda_i(w) \\
&\quad -\sum_{P_g^0(\epsilon)} \Psi_i^g(g_i(w) + \frac{1}{2}d_h^T\nabla^2 g_i(w)d_h) \\
&\quad -\sum_{P_=^0(\epsilon)} \Psi_i^{g-\lambda}((g_i(w) - \lambda_i(w)) + \frac{1}{2}d_h^T\nabla^2 g_i(w)d_h) \\
&\quad -\sum_{M^0(\epsilon)} \Psi_i^c(c_i(w) + \frac{1}{2}d_h^T\nabla^2 c_i(w)d_h) + o(\|d_h\|^2),
\end{aligned}
$$

$$
(d_h+d_v)^TB(d_h+d_v) = \frac{1}{2}d_h^TBd_h + d_h^TBd_v + \frac{1}{2}d_v^TBd_v.
$$

The above information, along with the definition of H, is now all combined in the following.

$$
\begin{aligned}
p_\mu(w+d_h+d_v) - p_\mu(w) &= -\gamma_Z^T H_Z^{-1}\gamma_Z - \sum_{T^0(\epsilon)} \Psi_i^G G_i(w) - \sum_{P_\lambda^0(\epsilon)} \Psi_i^\lambda \lambda_i(w) \\
&\quad -\sum_{P_g^0(\epsilon)} \Psi_i^g g_i(w) - \sum_{P_=^-(\epsilon)} \Psi_i^{g-\lambda}(g_i(w) - \lambda_i(w)) \\
&\quad -\sum_{M^0(\epsilon)} \Psi_i^c c_i(w) + \frac{1}{2}d_h^T H d_h \\
&\quad +d_h^TBd_v + \frac{1}{2}d_v^TBd_v + o(\|d_h\|^2) + o(\|d_h+d_v\|^2) \\
&\quad -\mu \sum_{T^0(\epsilon)} (\min(0, G_i(w+d_h+d_v)) - \min(0, G_i(w)))
\end{aligned}
$$

$$
\begin{aligned}
&+\mu \sum_{P^0_\lambda(\epsilon)} (|\lambda_i(w+d_h+d_v)| - |\lambda_i(w)|) \\
&+\mu \sum_{P^0_g(\epsilon)} (|g_i(w+d_h+d_v)| - |g_i(w)|) \\
&-\mu \sum_{P^-_=(\epsilon)} (\min(0, g_i(w+d_h+d_v) - \lambda_i(w+d_h+d_v)) \\
&\qquad - \min(0, g_i(w) - \lambda_i(w))) \\
&+\mu \sum_{M^0(\epsilon)} (|c_i(w+d_h+d_v)| - |c_i(w)|).
\end{aligned}
$$

Next, examine the changes in activities in more detail. Before looking at the individual cases, note the following.

$$
\begin{aligned}
\mathcal{A}^T d_h &= -\mathcal{A}^T Z(Z^T Z)^{-1} Z^T \gamma \;=\; 0 \\
\mathcal{A}^T d_v &= -\mathcal{A}^T \mathcal{A}(\mathcal{A}^T \mathcal{A})^{-1} \Phi(w+d_h) \\
&= -\Phi(w+d_h) \\
&= -\Phi(w) - d_h^T \mathcal{A} - \frac{1}{2} d_h^T \nabla^2 \Phi(w) d_h + o(\|d_h\|^2) \\
&= -\Phi(w) - \frac{1}{2} d_h^T \nabla^2 \Phi(w) d_h + o(\|d_h\|^2).
\end{aligned}
$$

- $i \in T^0(\epsilon)$

$$
\begin{aligned}
G_i(w+d_h+d_v) &= G_i(w) + (d_h+d_v)^T \nabla G_i(w) + \frac{1}{2}(d_h+d_v)^T \nabla^2 G_i(w)(d_h+d_v) \\
&\quad + o(\|d_h+d_v\|^2) \\
&= G_i(w) + d_h^T \nabla G_i(w) + d_v^T \nabla G_i(w) + \frac{1}{2} d_h^T \nabla^2 G_i(w) d_h \\
&\quad + d_h^T \nabla^2 G_i(w) d_v + \frac{1}{2} d_v^T \nabla^2 d_v + o(\|d_h+d_v\|^2) \\
&= G_i(w) + 0 - G_i(w) - \frac{1}{2} d_h^T \nabla^2 G_i(w) d_h + \frac{1}{2} d_h^T \nabla^2 G_i(w) d_h \\
&\quad + d_h^T \nabla^2 G_i(w) d_v + \frac{1}{2} d_v^T \nabla^2 G_i(w) d_v \\
&\quad + o(\|d_h+d_v\|^2) + o(\|d_h\|^2) \\
&= d_h^T \nabla^2 G_i(w) d_v + \frac{1}{2} d_v^T \nabla^2 G_i(w) d_v + o(\|d_h+d_v\|^2) + o(\|d_h\|^2).
\end{aligned}
$$

Let $\varsigma_i = \text{neg}[G_i(w)]$. Now, we have that

$$\begin{aligned}
&\sum_{T^0(\epsilon)} \min(0, G_i(w)) - \min(0, G_i(w + d_h + d_v)) \\
\leq\; & \sum_{T^0(\epsilon)} \varsigma_i G_i(w) + |d_h^T \nabla^2 G_i(w) d_v + \frac{1}{2} d_v^T \nabla^2 G_i(w) d_v| + o(\|d_h + d_v\|^2) + o(\|d_h\|^2) \\
\leq\; & \sum_{T^0(\epsilon)} \varsigma_i G_i(w) + \sum_{T^0(\epsilon)} |d_h^T \nabla^2 G_i(w) d_v| + \frac{1}{2} \sum_{T^0(\epsilon)} |d_v^T \nabla^2 G_i(w) d_v| + o(\|d_h + d_v\|^2) \\
& + o(\|d_h\|^2).
\end{aligned}$$

- $i \in P_\lambda^0(\epsilon)$

$$\begin{aligned}
|\lambda_i(w + d_h + d_v)| &= |\lambda_i(w) + d_h^T \nabla \lambda_i(w) + d_v^T \nabla \lambda_i(w) + 0| \\
&= |\lambda_i(w) + 0 - \lambda_i(w)| \\
&= 0.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{P_\lambda^0(\epsilon)} |\lambda_i(w + d_h + d_v)| - |\lambda_i(w)| &= \sum_{P_\lambda^0(\epsilon)} -|\lambda_i(w)| \\
&= -\sum_{P_\lambda^0(\epsilon)} \varrho_i \lambda_i(w),
\end{aligned}$$

where $\varrho_i = \text{sign}[\lambda_i(w)]$.

- $i \in P_g^0(\epsilon)$,

$$\begin{aligned}
&\sum_{P_g^0(\epsilon)} |g_i(w + d_h + d_v)| - |g_i(w)| \\
\leq\; & \sum_{P_g^0(\epsilon)} |d_h^T \nabla^2 g_i(w) d_v| + \frac{1}{2} \sum_{P_g^0(\epsilon)} |d_v^T \nabla^2 g_i(w) d_v| - \sum_{P_g^0(\epsilon)} \varrho_i g_i(w) \\
& + o(\|d_h + d_v\|^2) + o(\|d_h\|^2),
\end{aligned}$$

where $\varrho_i = \text{sign}[g_i(w)]$.

- $i \in P_{=}^{-}(\epsilon)$, let $\varsigma_i = \text{neg}[g_i(w) - \lambda_i(w)]$.

$$\begin{aligned}
&\sum_{P_{=}^{-}(\epsilon)} \min(0, g_i(w) - \lambda_i(w)) - \min(0, g_i(w + d_h + d_v) - \lambda_i(w + d_h + d_v)) \\
\leq\ & \sum_{P_{=}^{-}(\epsilon)} \varsigma_i(g_i(w) - \lambda_i(w)) + |d_h^T \nabla^2 g_i(w) d_v + \frac{1}{2} d_v^T \nabla^2 g_i(w) d_v| \\
&+ o(\|d_h + d_v\|^2) + o(\|d_h\|^2) \\
\leq\ & \sum_{P_{=}^{-}(\epsilon)} \varsigma_i(g_i(w) - \lambda_i(w)) + \sum_{P_{=}^{-}(\epsilon)} |d_h^T \nabla^2 g_i(w) d_v| + \frac{1}{2} \sum_{P_{=}^{-}(\epsilon)} |d_v^T \nabla^2 g_i(w) d_v| \\
&+ o(\|d_h + d_v\|^2) + o(\|d_h\|^2).
\end{aligned}$$

- $i \in M^0(\epsilon)$

$$\begin{aligned}
&\sum_{M^0(\epsilon)} |c_i(w + d_h + d_v)| - |c_i(w)| \\
\leq\ & \sum_{M^0(\epsilon)} |d_h^T \nabla^2 c_i(w) d_v| + \frac{1}{2} \sum_{M^0(\epsilon)} |d_v^T \nabla^2 c_i(w) d_v| - \sum_{M^0(\epsilon)} \varrho_i c_i(w) + \\
&+ o(\|d_h + d_v\|^2) + o(\|d_h\|^2),
\end{aligned}$$

where $\varrho_i = \text{sign}[c_i(w)]$.

Combining all the above information, and grouping similar terms we get the following result.

$$\begin{aligned}
p_\mu(w + d_h + d_v) - p_\mu(w) \ =\ & -\gamma_Z^T H^{-1} \gamma_Z - \sum_{T^0(\epsilon)} (\Psi_i^G - \mu\varsigma_i) G_i(w) \\
& - \sum_{P_\lambda^0(\epsilon)} (\Psi_i^\lambda + \mu\varrho_i) \lambda_i(w) - \sum_{P_g^0(\epsilon)} (\Psi_i^g + \mu\varrho_i) g_i(w) \\
& - \sum_{P_{=}^{-}(\epsilon)} (\Psi_i^{g-\lambda} - \mu\varsigma_i)(g_i(w) - \lambda_i(w)) \\
& - \sum_{M^0(\epsilon)} (\Psi_i^c + \mu\varrho_i) c_i(w) \\
& + d_h^T B d_v + \mu \sum_{T^0(\epsilon)} |d_h^T \nabla^2 G_i(w) d_v| + \mu \sum_{P_g^0(\epsilon)} |d_h^T \nabla^2 g_i(w) d_v|
\end{aligned}$$

$$+\mu \sum_{P_{=}^{-}(\epsilon)} |d_h^T \nabla^2 g_i(w) d_v| + \mu \sum_{M^0(\epsilon)} |d_h^T \nabla^2 c_i(w) d_v|$$
$$+\frac{1}{2}\left(d_v^T B d_v + \mu \sum_{T^0(\epsilon)} |d_v^T \nabla^2 G_i(w) d_v| \right.$$
$$+\mu \sum_{P_g^0(\epsilon)} |d_v^T \nabla^2 g_i(w) d_v| + \mu \sum_{P_{=}^{-}(\epsilon)} |d_v^T \nabla^2 g_i(w) d_v|$$
$$\left. +\mu \sum_{M^0(\epsilon)} |d_v^T \nabla^2 c_i(w) d_v| \right) + o(\|d_h + d_v\|^2) + o(\|d_h\|^2).$$

For any of the activities a_i, let $r_i^a = d_h^T \nabla^2 a_i(w) d_h$. Then, using this definition, we can write

$$\Phi(w + d_h) = \Phi(w) + \frac{1}{2} r + o(\|d_h\|^2)$$

and

$$d_v = -\mathcal{A}(\mathcal{A}^T \mathcal{A})^{-1}[\Phi + \frac{1}{2} r] + d(\|d_h\|^2).$$

Define $H_1 = B\mathcal{A}(\mathcal{A}^T\mathcal{A})^{-1}$ and $H_2 = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T H_1$. Therefore,

$$\begin{aligned}
d_v^T B d_v &= (-\mathcal{A}(\mathcal{A}^T\mathcal{A})^{-1}[\Phi + \frac{1}{2}r])^T B(-\mathcal{A}(\mathcal{A}^T\mathcal{A})^{-1}[\Phi + \frac{1}{2}r]) + o(\|d_h\|^2) \\
&= (\frac{1}{2}r^T + \Phi^T) H_2 (\Phi + \frac{1}{2}r) + o(\|d_h\|^2) \\
&= \Phi^T H_2 \Phi + \Phi^T H_2 r + \frac{1}{4} r^T H_2 r + o(\|d_h\|^2) \\
&= \Phi^T H_2(\Phi + r) + \frac{1}{4} r^T H_2 r + o(\|d_h\|^2) \\
&= \Phi^T t + \frac{1}{4} r^T H_2 r + o(\|d_h\|^2),
\end{aligned}$$

where $t = H_2(\Phi + r)$. Since $r^T H_2 r$ is $o(\|d_h\|^4)$, it can be included in the $o(\|d_h\|^2)$ term. Therefore,

$$\begin{aligned}
\frac{1}{2}|d_v^T B d_v| &\leq |\Phi^T t| + o(\|d_h\|^2) \\
&\leq \frac{1}{2} \sum |a_i(w)| \cdot |t_i(w)| + o(\|d_h\|^2),
\end{aligned}$$

where the sums are over all the activities.

Define $\bar{H}_i = \nabla^2 a_i \mathcal{A}(\mathcal{A}^T\mathcal{A})^{-1}$ and $\tilde{H}_i = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}\bar{H}_i$ for each activity. Let

$$u^j(w) = \tilde{H}_j[\Phi(w) + r], \text{ and } \bar{u}_i = \sum_j |u_i^j(w)|.$$

Therefore,

$$\frac{1}{2}\sum |d_v^T \nabla^2 a_i d_v| \leq \frac{1}{2}\sum_i |a_i(w)| \cdot |\bar{u}_i(w)| + o(\|d_h\|^2).$$

Using a similar technique, it can be verified that

$$\begin{aligned} |d_h^T B d_v| &\leq \sum_i |y_i(w)| \cdot |a_i(w)| + o(\|d_h\|^2) \\ \sum_i |d_v^T \nabla^2 d_v| &\leq \sum_i |a_i(w)| \cdot |\bar{s}_i(w)| + o(\|d_h\|^2) \end{aligned}$$

where $y^T = d_h^T H_1$, $s^j(w) = d_h^T \tilde{H}_j$ and $\bar{s}_i(w) = \sum_j |s_i^j(w)|$.

So, we can now write

$$\begin{aligned} & p_\mu(w + d_h + d_v) - p_\mu(w) \\ \leq\ & -\gamma_Z^T H^{-1}\gamma_Z - \sum_{T^0(\epsilon)} (\Psi_i^G - \mu\varsigma_i) G_i(w) - \sum_{P_\lambda^0(\epsilon)} (\Psi_i^\lambda + \mu\varrho_i)\lambda_i(w) \\ & - \sum_{P_g^0(\epsilon)} (\Psi_i^g + \mu\varrho_i) g_i(w) - \sum_{P_=^-(\epsilon)} (\Psi_i^{g-\lambda} - \mu\varsigma_i)(g_i(w) - \lambda_i(w)) \\ & - \sum_{M^0(\epsilon)} (\Psi_i^c + \mu\varrho_i) c_i(w) \\ & + \sum_i |y_i(w)| \cdot |a_i(w)| + \mu \sum_i |a_i(w)| \cdot |\bar{s}_i(w)| + \frac{1}{2}\sum_i |a_i(w)| \cdot |t_i(w)| \\ & + \frac{1}{2}\mu \sum_i |a_i(w)| \cdot |\bar{u}_i(w)| + o(\|d_h + d_v\|^2) + o(\|d_h\|^2). \end{aligned}$$

For each activity a_i, define the function

$$\xi_i = |y_i| + \mu|\bar{s}_i| + \frac{1}{2}|t_i| + \frac{1}{2}\mu|\bar{u}_i|.$$

Now,

$$
\begin{aligned}
p_\mu(w+d_h+d_v)-p_\mu(w) \;=\;& -\gamma_Z^T H_Z^{-1}\gamma_Z \\
& + \sum_{T^0(\epsilon)} (\xi_i(w)|G_i(w)| - (\Psi_i^G - \mu\varsigma_i)G_i(w)) \\
& + \sum_{P_\lambda^0(\epsilon)} (\xi_i(w)|\lambda_i(w)| - (\Psi_i^\lambda + \mu\varrho_i)\lambda_i(w)) \\
& + \sum_{P_g^0(\epsilon)} (\xi_i(w)|g_i(w)| - (\Psi_i^g + \mu\varsigma_i)g_i(w)) \\
& + \sum_{P_=^-(\epsilon)} (\xi_i(w)|g_i(w)-\lambda_i(w)| - \\
& \qquad (\Psi_i^{g-\lambda} - \mu\varsigma_i)(g_i(w)-\lambda_i(w))) \\
& + \sum_{M^0(\epsilon)} (\xi_i(w)|c_i(w)| - (\Psi_i^c + \mu\varrho_i)c_i(w)) \\
& + o(\|d_h+d_v\|^2) + o(\|d_h\|^2).
\end{aligned}
$$

Returning to the activities, we get the following results.

- For $i \in T^0(\epsilon)$, if $G_i(w) < 0$, then

$$-(\Psi_i^G - \mu\varsigma_i)G_i(w) = (\mu - \Psi_i^G)G_i(w) = -|\mu - \Psi_i^G| \cdot |G_i(w)|.$$

  If $G_i(w) > 0$, then

$$-(\Psi_i^G - \mu\varsigma_i)G_i(w) = -\Psi_i^G G_i(w) = -|\Psi_i^G| \cdot |G_i(w)|.$$

  Therefore, in either case,

$$-(\Psi_i^G - \mu\varsigma_i)G_i(w) = -|\Psi_i^G - \mu\varsigma_i| \cdot |G_i(w)|.$$

- For $i \in P_\lambda^0(\epsilon)$, as above, it can be shown that

$$-(\Psi_i^\lambda + \mu\varrho_i)\lambda_i = -|\Psi_i^\lambda + \mu\varrho_i| \cdot |\lambda_i(w)|.$$

- For $i \in P_g^0(\epsilon)$,

$$-(\Psi_i^g + \mu\varrho_i)g_i = -|\Psi_i^g + \mu\varrho_i| \cdot |g_i(w)|.$$

- For $i \in P_{=}^{-}(\epsilon)$,

$$-(\Psi_i^{g-\lambda} - \mu\varsigma_i)(g_i(w) - \lambda_i(w)) = -|\Psi_i^{g-\lambda} - \mu\varsigma_i| \cdot |g_i(w) - \lambda_i(w)|.$$

- For $i \in M^0(\epsilon)$,

$$-(\Psi_i^c + \mu\varrho_i)\lambda = -|\Psi_i^c + \mu\varrho_i| \cdot |c_i(w)|.$$

Rewriting the difference in the penalty function, we get

$$\begin{aligned} p_\mu(w + d_h + d_v) - p_\mu(w) &= -\gamma_Z H_Z^{-1} \gamma_Z + \zeta(w) \\ &\quad o(\|d_h + d_v\|^2) + o(\|d_h\|^2), \end{aligned}$$

where

$$\begin{aligned} \zeta(w) &= \sum_{T^0(\epsilon)} (\xi_i(w) - |\Psi_i^G - \mu\varsigma_i|)|G_i(w)| \\ &\quad + \sum_{P_\lambda^0(\epsilon)} (\xi_i(w) - |\Psi_i^\lambda + \mu\varrho_i|)|\lambda_i(w)| \\ &\quad + \sum_{P_g^0(\epsilon)} (\xi_i(w) - |\Psi_i^g + \mu\varrho_i|)|g_i(w)| \\ &\quad + \sum_{P_{=}^{-}(\epsilon)} (\xi_i(w) - |\Psi_i^{g-\lambda} - \mu\varsigma_i|)|g_i(w) - \lambda_i(w)| \\ &\quad + \sum_{M^0(\epsilon)} (\xi_i(w) - |\Psi_i^c + \mu\varrho_i|)|c_i(w)|. \end{aligned}$$

Now, as $w \to \hat{w}$, $\xi_i(w) \to 0$. Also $\Psi_i - \mu\varsigma_i \neq 0$ and $\Psi_i + \mu\varrho_i \neq 0$ because $\hat{w}$ is a strict second order point. Therefore, there exists $\hat{\delta}_\Phi > 0$ such that when "close enough",

$$\zeta(w) \leq -\hat{\delta}_\Phi \|\Phi(w)\|_1.$$

Therefore,

$$p_\mu(w + d_h + d_v) - p_\mu(w) \leq -\gamma_Z^T H_Z^{-1} \gamma_Z - \hat{\delta}_\Phi \|\Phi(w)\|_1 + o(\|d_h + d_v\|^2) + o(\|d_h\|^2).$$

When "close enough", the norm of H_Z is bounded above and below by constants. From Assumption 5.13,

$$\gamma_Z^T H_Z^{-1} \gamma_Z \geq b_1^H \|\gamma_Z\|^2 = b_1^H \|Z^T\gamma\|^2.$$

Combining everything, we have that

$$p_\mu(w + d_h + d_v) - p_\mu(w) \quad \leq \quad -b_1^H \|Z^T\gamma\|_2^2 - \hat{\delta}_\Phi \|\Phi(w)\|_1 + o(\|d_h + d_v\|^2) + o(\|d_h\|^2).$$

Examining the lengths of the horizontal and vertical steps, and using Assumption 5.13,

$$\|d_h\|_2^2 = d_h^T d_h = \gamma_Z^T (H_Z^{-1})^T (H_Z^{-1}) \gamma_Z \leq \|\gamma_Z\|_2^2 \|H_Z^{-1}\|_2^2 \leq \delta_{HZ}^2 \|Z^T\gamma\|_2^2$$

and

$$\|d_h + d_v\|_2^2 = \|d_h\|_2^2 + \|d_v\|_2^2 \leq \delta_{HZ}^2 \|Z^T\gamma\|_2^2 + \|d_v\|_2^2.$$

Now, there exists $\delta_r > 0$ such that

$$\begin{aligned}
\|d_v\|_2^2 &= (\Phi + \frac{1}{2}r)^T (\mathcal{A}^T\mathcal{A})^{-1} (\Phi + \frac{1}{2}r) + o(\|d_h\|^4) \\
&\leq \|\Phi + \frac{1}{2}r\|^2 \|\mathcal{A}^T\mathcal{A}\| + o(\|d_h\|_4) \\
&\leq \delta_\mathcal{A} \|\Phi\|^2 + \frac{1}{2}r\|^2 + o(\|d_h\|^4) \\
&\leq \delta_\mathcal{A} \|\Phi\|^2 + o(\|d_h\|^3) \\
&\leq \delta_\mathcal{A} \|\Phi\|^2 + \delta_r \|d_h\|^2 \\
&\leq \delta_\mathcal{A} \|\Phi\|^2 + \delta_r \|Z^T\gamma\|^2.
\end{aligned}$$

Therefore, $o(\|d_h + d_v\|^2) + o(\|d_h\|^2) \leq \delta_\mathcal{A} \|\Phi\|^2 + \delta_{r_1} \|Z^T\gamma\|^2$, for some positive constants δ_{r_1}, and $\delta_\mathcal{A}$. We can assume that $\|\Phi\| \leq \|\Phi\|_1$ because Φ contains the values of the ϵ-activities.

Therefore, when close enough, there exists positive constants L_1 and L_2 such that

$$\begin{aligned}
p_\mu(w + d_h + d_v) - p_\mu(w) &\leq -(L_1 \|Z^T\gamma\|_2^2 + L_2 \|\Phi\|_1) \\
&= -\min(L_1, L_2)(\|Z^T\gamma\|_2^2 + \|\Phi\|_1).
\end{aligned}$$

Defining $L = \min(L_1, L_2)$ gives the desired result. □

Corollary 5.41 *Given Assumptions 5.1 - 5.13, if a type four point w is "close enough" to $\hat{w}$, then $p_\mu(w + d_h + d_v) < p_\mu(w)$.*

Proof: This follows immediately from the above lemma, and the fact that if w is not itself a strict second order point, then either $||\Phi|| \neq 0$ or $||\gamma_Z|| \neq 0$. □

Lemma 5.42 *Given Assumptions 5.1 - 5.13, if the iterate w is close enough to $\hat{w}$, then $d_N = d_h + d_v$ will eventually be accepted by the algorithm.*

Proof: To prove this result, we need to establish the following points.

- The Newton step eventually provides sufficient decrease in the penalty function.

  From Corollary 5.41, d_N eventually decreases the penalty function. Therefore, it remains to show that this decrease is sufficient, in the sense described in Section 4.2. For sufficient decrease, it is required that

$$p_\mu(w + d_N) - p_\mu(w) \leq r_1 d_N^T \gamma.$$

  Now, looking at the right hand side in the same way that the proof above was established, we have that

$$\begin{aligned} d_N^T\gamma &= d_h^T\gamma + d_v^T\gamma \\ &\geq -\gamma_Z^T H_Z \gamma_Z - \sum \Psi_i^a a_i(w) + \frac{1}{2} d_h^T \gamma_Z \\ &= -\frac{1}{2}\gamma_Z^T H_Z^T \gamma_Z - \sum \Psi_i^a a_i(w) \\ &\geq -\frac{1}{2} b_2^H ||\gamma_Z||_2^2 - \mu||\Phi(w)||_1 \\ &= -M(||Z^T\gamma||_2^2 + ||\Phi||_1) \end{aligned}$$

for a positive constant M. As long as $L \geq r_1 M$ (which is reasonable since r_1 is assigned a small value in our implementation), it follows that

$$\begin{aligned} p_\mu(w + d_N) - p_\mu(w) &\leq -L(\|Z^T\gamma\|_2^2 + \|\Phi\|_1) \\ &\leq -r_1 M(\|Z^T\gamma\|_2^2 + \|\Phi\|_1) \\ &\leq -r_1 d_N^T \gamma. \end{aligned}$$

Therefore, d_N will eventually provide sufficient decrease in the penalty function.

- The Newton step eventually falls within the trust region.

  As the iterates are approaching a second order point, the Newton direction is decreasing in size. Recall that near a solution, a quadratic model is a good predictor of the penalty function, and the trust region radius will not be reduced, although it may be increased. Therefore, eventually, $\|d_N\|_\infty \leq \Delta$ will be satisfied.

Therefore, a Newton step will be accepted by the algorithm. □

5.5.3 Decreases of the Algorithm Tolerances

It will now be proven that the activity tolerance ϵ and the closeness tolerance Λ are only decreased within the algorithm a finite number of times. A similar result was presented in parts 5 and 6 of Theorem 1 in [34]. An alternate way of stating this result is that ϵ and Λ are bounded below by some positive constants, or that

$$\lim_{k\to\infty} \epsilon^k \not\to 0 \text{ and } \lim_{k\to\infty} \Lambda^k \not\to 0.$$

Lemma 5.43 *Given Assumptions 5.1 - 5.13, the sequence of iterations produced by the algorithm must satisfy*

$$\lim_{k\to\infty} \epsilon^k \not\to 0 \textit{ and } \lim_{k\to\infty} \Lambda^k \not\to 0.$$

Proof: The proof will proceed by contradiction. Assume that

$$\lim_{k\to\infty} \epsilon^k \to 0.$$

Since Λ^k is decreased whenever ϵ^k is decreased, it follows that

$$\lim_{k\to\infty} \Lambda^k \to 0.$$

Assume that

$$\lim_{k\to\infty} \|(Z^k)^T \gamma^k\| \not\to 0,$$

i.e. that there exists some constant $\delta_Z > 0$ such that $\|(Z^k)^T\gamma^k\| > \delta_Z$. This implies that beyond some point, all the iterates are type one points. However, in this situation ϵ and Λ are not decreased. Therefore the assumption was incorrect, and

$$\lim_{k\to\infty} \|(Z^k)^T \gamma^k\| \to 0.$$

Since $\|(Z^k)^T\gamma^k\|$ is approaching zero, the iterates are approaching a stationary point. Now, in stage two of the convergence proof, it was established that only a finite number of stationary, non-first order points exist. Therefore, it follows that in the limit, the iterates must approach a stationary point which is a first order point, and hence a strict second order point. As well, the algorithm will identify the correct set of activities, and will attempt the full Newton step. It was just established above that such steps will be successful when close enough to the strict second order point $\hat{w}$. Therefore, ϵ^k will not be repeatedly reduced, and the assumption that it tends to zero is incorrect. □

5.5.4 Behavior Near a Second Order Point

Lemma 5.44 *Given Assumptions 5.1 - 5.13, if ϵ^k does not tend to zero, then eventually all the iterations are successful Newton steps.*

Proof: We have already established that only a finite number of successful steps can be taken from type one or type two points, and that the algorithm will eventually approach a stationary point. From earlier discussions, it follows that the final stationary point approached must be a second order point. Since ϵ^k is not repeatedly reduced, full Newton steps are taken until convergence. □

5.5.5 Convergence of the Algorithm

Corollary 5.45 *Given Assumptions 5.1 - 5.13, the algorithm will converge to a strict second order point of the penalty function.*

Proof: When very close to a second order point, the algorithm has correctly identified the active set, and is essentially Newton's algorithm applied to an unconstrained minimization problem. Therefore, convergence to a strict second order point will be observed. □

Chapter 6

Degeneracy in the Penalty Function

6.1 Introduction

In previous chapters, we have assumed that all points encountered by the algorithm were nondegenerate. However, in practice, the use of ϵ-activities may result in a large number of penalty terms being considered active. Consequently, there is an increased possibility of linear dependencies among the gradients of the active penalty terms, that is, that some points will be degenerate. Of course, even for exact activities, degeneracy is possible.

In this chapter, we discuss some of the problems caused by degeneracy. Several traditional techniques for resolving degeneracy in one level mathematical programs are summarized, along with some difficulties encountered when applying those techniques to the bilevel problem penalty function. Finally, a new technique for resolving degeneracy in our problem is described and proven to work for exact activities. An extended version of the technique for ϵ-activities is used within the implemented version of the algorithm.

6.2 Problems Caused by Degeneracy

Degeneracy refers to the situation in which the gradients of the active terms at the current point are not linearly independent. When the current point appears to be far from a stationary point, the trust region direction is chosen to maintain all activities up to first order change. The presence of degeneracy does not alter this stage of the algorithm in any way. In effect, the degeneracy is irrelevant. However, when the current point appears to be close to a stationary point, it is necessary to estimate the values of the Lagrangian multipliers by solving, in a least squares sense, the system of equations

$$\mathcal{A}\Psi = \gamma,$$

where all terms are evaluated at the current point. At a degenerate point, $\mathcal{A}$ is not full column rank, so the system may not have a unique solution. Corresponding to each basic subset of the columns of $\mathcal{A}$, there is a unique set of multipliers. However, there may be an exponential number of choices of basic subsets. This nonuniqueness of the activities causes several problems in our algorithm.

- The numerous choices of bases may lead to an incorrect classification of the status of the current point. For example, if a particular set of multipliers includes out of kilter values, then a dropping direction is indicated. However, there may exist a choice of basis whose multipliers are all in kilter, which suggests that all the activities should be maintained.

- Even if the point is correctly classified as being near a stationary non-first order point, the nonuniqueness of the sets of multipliers may lead to attempting to drop some activities which are actually active at a solution.

- Changes in the activities whose gradients are in the current basis may result in changes in some of the nonbasic activities as well. Therefore, a dropping direction defined from the current basis may not be a descent direction for the entire penalty function.

Note that it is possible to correctly classify a dropping situation in the presence of degeneracy, and then to define a dropping direction which provides descent. However, the presence of degeneracy complicates the decision making process.

6.3 Traditional Degeneracy Resolving Techniques

Consider solving the linear optimization problem

$$LP: \min_{x \in \mathbb{R}^n} c^T x \text{ s.t. } G^T x \geq f.$$

The necessary conditions at a nondegenerate solution of LP include the existence of a unique set of multipliers $\lambda \geq 0$ satisfying $A\lambda = c$, where A, the activity matrix at x, contains a subset of the columns of G.

These necessary conditions can be extended to the case of degeneracy. The uniqueness requirement for λ is replaced by the requirement that there exists at least one nonnegative solution λ to the underdetermined system $A\lambda = c$. An obvious technique to resolve the uncertainties caused by degeneracy is to examine all basic subsets of A to determine optimality or a dropping direction. The following presents this algorithmic framework.

Algorithm 6.1 (Traditional Degeneracy Resolving Algorithm)

- *REPEAT*

  1. *Partition the columns of A to form two matrices A_B and A_N which satisfy Range$[A_B]$ = Range$[A]$ and $[A_B, A_N] = A$.*

2. *Solve* $A_B\lambda_B = c$ *for* λ_B.

3. *Set* $\lambda_N = 0$.

- *UNTIL* $\lambda^T = (\lambda_B^T, \lambda_N^T)$ *is in kilter*

- *OR UNTIL a feasible descent direction for LP is identified.*

Given the matrix A, there may be an exponential number of choices of A_B. Therefore, rather than enumerating all possible bases, most degeneracy resolving algorithms attempt to efficiently search the possibilities. Two different techniques for generating matrices A_B are described below.

6.3.1 Perturbation

The perturbation technique for resolving degeneracy in LP involves transforming, or perturbing, the degenerate problem LP into a closely related nondegenerate problem LP(ε), for some vector ε having small positive components. For example, consider the constraints

$$\begin{aligned} C1: \; b_1^T x - f_1 &\geq 0 \\ C2: \; b_2^T x - f_2 &\geq 0 \\ C3: \; b_3^T x - f_3 &\geq 0 \\ C4: \; b_4^T x - f_4 &\geq 0, \end{aligned}$$

where the first three constraints are active at a common, degenerate point, denoted D1 in Figure 6.1. This feasible region is slightly modified to

$$\begin{aligned} C1': \; b_1^T x - f_1 - \varepsilon_1 &\geq 0 \\ C2': \; b_2^T x - f_2 - \varepsilon_2 &\geq 0 \\ C3': \; b_3^T x - f_3 - \varepsilon_3 &\geq 0 \\ C4': \; b_4^T x - f_4 - \varepsilon_4 &\geq 0, \end{aligned}$$

where ε_1, ε_2, ε_3, and ε_4 are very small positive values, on the order or magnitude of ten times machine epsilon. The very small values are required so that the feasible regions of LP and LP(ϵ) are closely related.

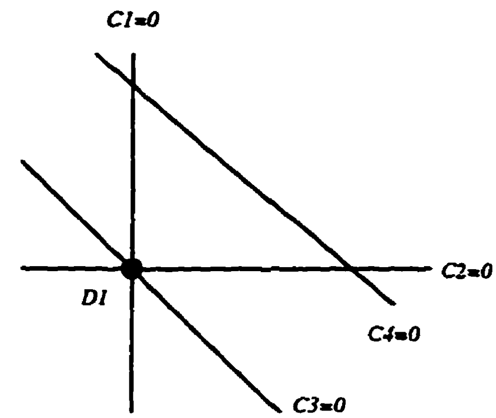

Figure 6.1: Original Feasible Region

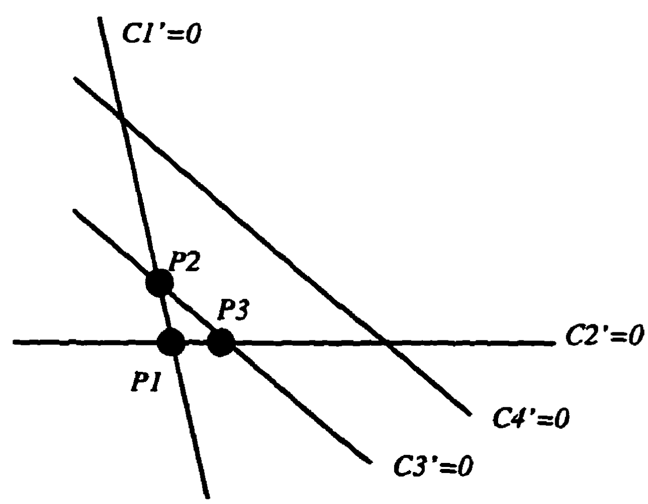

Figure 6.2: Perturbed Feasible Region

The degenerate vertex D1 has been transformed into three nondegenerate points P1, P2, and P3, as shown in Figure 6.2. The activity matrices at these points correspond to the possible basis matrices at D1 in the original problem. If the linear program algorithm determines that the multipliers at either of the perturbed vertices satisfy the necessary optimality conditions for LP(ε), then D1 satisfies the necessary conditions for a solution of LP. However, if any of the other vertices of the perturbed problem are reached, that is, if the fourth constraint is reached from P1, P2, or P3, then degeneracy has been resolved, and the linear program algorithm can proceed from the corresponding point in the unperturbed problem.

6.3.2 Ryan-Osborne Approach

In [61], Ryan and Osborne study the issue of resolving degeneracy in linear programs of the form

$$\text{LP1: } \min_{x \in \mathbb{R}^n} c^T x \text{ subject to } A^T x = f \text{ and } x \geq 0.$$

A point is optimal for this problem if there exists $\lambda \geq 0$ and Ψ such that

$$A\Psi + \mathcal{J}^0 \lambda = c,$$

where $\mathcal{J}^0$ is a submatrix of the identity matrix corresponding to elements of x which have value zero at the current point. Note that there is no restriction on the value of the Ψ multipliers because these constraints must remain active for feasibility.

The matrix A is divided into a basic submatrix A_B and a nonbasic submatrix A_N. Correspondingly, the variables are divided into a basic index set $\mathcal{S}_B$ and a nonbasic index set $\mathcal{S}_N$. Let x_B and x_N denote the basic and nonbasic variables. Note that $x_N = 0$ is always satisfied. Let

$$\mathcal{S}_B^0 = \{i \in \mathcal{S}_B : x_i = 0\}.$$

The current point is degenerate if $\mathcal{S}_B^0 \neq \emptyset$.

Ryan and Osborne define a subproblem of LP1 which is used to resolve degeneracy at x. The inactive and nondegenerate part of the problem can be ignored because degeneracy is a local issue. Therefore, the authors describe the subproblem SP1.

$$\text{SP1}: \min_{x \in \mathbb{R}^n} c^T x \quad \text{subject to} \quad A^T x = f \text{ and } x_i \geq 0, i \in \mathcal{S}_N \cup \mathcal{S}_B^0.$$

If there exists a direction of recession, that is, a direction of unbounded feasible descent, for SP1 from x, then this direction is also a feasible descent direction for LP1 from x. Meanwhile, if the current point is optimal for SP1, then it is optimal for the original problem. However, the current point is a degenerate point for SP1 as well as LP1.

A direction d is a direction of recession for SP1 from x if and only if it satisfies the following conditions:

$$c^T d < 0, \; A^T d = 0, \text{ and } d_i \geq 0 \text{ for } i \in \mathcal{S}_N \cup \mathcal{S}_B^0.$$

These conditions are independent of the vector b and the value of x. Therefore, the authors describe another problem similar to SP1, but using a set of random values. Let

$$x_i^r = \begin{cases} r_i & \text{if } i \in \mathcal{S}_B^0 \\ x_i & \text{otherwise,} \end{cases}$$

where each r_i can take any positive value. Now, consider the problem

$$\text{SP2}: \min_{y \in \mathbb{R}^n} c^T y \quad \text{subject to} \quad A^T y = f^r \text{ and } y_i \geq 0, i \in \mathcal{S}_N \cup \mathcal{S}_B^0,$$

where $f^r = f + A^T r$. Note that $y = x^r$ is a feasible point for SP2.

Any direction d is a direction of recession for SP2 from y if and only if it is a direction of recession for SP1 from x, and therefore, a direction of feasible descent for LP1 from x. Conversely, if no direction of recession exists for SP2 starting from y, then an optimal solution, including an optimal set of multipliers, can be obtained for SP2. These multipliers correspond to a choice of basis for the original problem LP1 at x.

When solving SP2 starting from y, at least one nondegenerate step can be taken. A further level of degeneracy may be encountered in the solution process, depending on the random values used in defining SP2. If this situation arises, then the same process can be invoked recursively. Note that since an initial step is taken away from x^r, any other degenerate points encountered will have fewer activities, and the recursion process will be finite.

6.4 Examples of Degeneracy in the Penalty Function

The techniques described above for resolving degeneracy are based on the premise that the uniqueness requirement for multipliers in the necessary conditions at a nondegenerate point can be replaced by the existence of such multipliers at a degenerate point. The examples in this section show that a similar correspondence between necessary conditions at nondegenerate and degenerate solutions of the bilevel penalty function does not exist.

Consider example problem BP_2 from Chapter 2. With the penalty parameter μ fixed at one, the unconstrained function

$$\begin{aligned} p(x, y, \lambda) &= 2x + y - \min(0, x + 2y) + |x - y + \lambda_1 - \lambda_2 - 1| + \\ &\quad |\min(1 - 2x - y, \lambda_1)| + |\min(x + y + 1, \lambda_2)| \end{aligned}$$

is minimized. It can be shown that $p(x, y, \lambda) \geq 1$. Consider the following points.

- At the point $w_1 = (0, 0, 1, 0)$, the activity matrix is

$$\mathcal{A} = \left[\begin{array}{cccc} \nabla G_1 & \nabla c_1 & \nabla g_1 - \nabla\lambda_1 & \nabla\lambda_2 \\ \hline 1 & 1 & -2 & 0 \\ 2 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{array}\right],$$

where $P_{=}^{+} = \{1\} \neq \emptyset$. According to the nondegenerate necessary conditions, a dropping direction can always be defined in this case. However, w_1 is both degenerate (since rank[$\mathcal{A}$] = 3) and optimal for p (since $p(w_1) = 1$). Dropping $g_1 - \lambda_1 = 0$ increases some of the other active penalty terms so much that the penalty function is increased (even for very small steps) due to the linear dependencies among the gradients in $\mathcal{A}_B$. Therefore, the requirement that $P_{=}^{+} = \emptyset$ does not extend to the degenerate case.

- At the point $w_2 = (2/3, -1/3, 0, 0)$, the activity matrix,

$$\mathcal{A} = \left[\begin{array}{ccccc} \nabla G_1 & \nabla c_1 & \nabla g_1 & \nabla \lambda_1 & \nabla \lambda_2 \\ \hline 1 & 1 & -2 & 0 & 0 \\ 2 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{array}\right],$$

has five columns, but is of rank four. Any choice of four of the columns forms a basis for the full activity matrix. Consider the following:

  - If ∇g_1 is in the basis matrix (as it is for four of the five choices for $\mathcal{A}_B$), then the multipliers for that basis are $\Psi_1^g = -1$ and zero for all the other basic activities since $\nabla g_1 = -\gamma$. The nondegenerate optimal range for Ψ_1^g is $\Psi_1^g \in [0, 1]$, so these multipliers are not in kilter according to the rules derived assuming nondegeneracy.

  - If ∇g_1 is not in the basis, the only choice of basis is

$$\mathcal{A}_B = [\nabla G_1,\ \nabla c_1,\ \nabla \lambda_1,\ \nabla \lambda_2].$$

  The multipliers for this basis satisfy

$$\Psi = [\Psi_1^G,\ \Psi_1^c,\ \Psi_1^\lambda,\ \Psi_2^\lambda] = [1,\ 1,\ -1,\ 1].$$

The multiplier $\Psi_1^\lambda = -1$ violates the nondegenerate necessary condition that $\Psi_1^\lambda \in [0, 1]$.

Therefore, any choice of $\mathcal{A}_B$ at w_2 results in a violation of the necessary optimality conditions derived for nondegenerate points. However, this degenerate point is a minimum point of $p(x, y, \lambda)$ since $p(w_2) = 1$.

These two cases show that at a degenerate solution of the penalty function,

- the set $P_\pm^+$ need not be empty.
- there may not exist a basis of the activity matrix which defines a set of multipliers which are in kilter according to the necessary conditions derived assuming nondegeneracy.

6.5 Problems Applying the Traditional Techniques

The two degeneracy resolving techniques described previously depend on the existence of a basis matrix for the activities which defines a set of multipliers satisfying the necessary conditions derived assuming nondegeneracy. Therefore, they will not be appropriate for resolving degeneracy in our penalty function.

A problem which arises when applying the perturbation technique is that small changes to the constraints may significantly alter the set of necessary optimality conditions at a nondegenerate point. Consider the unperturbed situation $\lambda_i(w) = g_i(w) = 0$. The nondegenerate necessary conditions require that $0 \leq \Psi_i^\lambda \leq \mu$, $0 \leq \Psi_i^g \leq \mu$, and $\Psi_i^\lambda + \Psi_i^g \leq \mu$. If, as a result of some perturbation, $\lambda_i(w)$ is considered active and less than inactive $g_i(w)$ in the perturbed space, the necessary nondegenerate condition for optimality is $-\mu \leq \Psi_i^\lambda \leq \mu$. Clearly, there are values for the multipliers for which we would conclude that the perturbed problem satisfies the necessary nondegenerate conditions, while the unperturbed problem does not. The multipliers obtained from the perturbed problem may neither satisfy the nondegenerate necessary conditions nor define a descent direction for the penalty function.

When attempting to apply the Ryan-Osborne technique to resolving degeneracy in the penalty function, we were able to verify that a direction of recession for a linear problem defined at the degenerate point (perturbed or not) was a direction of descent for the penalty function at the degenerate point. However, it was not possible to equate the cases of finding a solution for the perturbed linear problem and the current point being a first order point of the penalty function.

A new approach is required to resolve degeneracy in the penalty function p_μ.

6.6 Resolving Degeneracy in the Penalty Function

To determine if descent from a degenerate point is possible, changes in the nonbasic activities must be considered in relation to changes in the basic activities.

Algorithm 6.2 (Framework for Resolving Degeneracy in p_μ)

1. *Construct a basic activity matrix $\mathcal{A}_B$ from the activity matrix $\mathcal{A}$:*
   - *Include $\nabla\lambda_i(w)$ for $i \in P^0_\lambda \cup P^0_=$, $\nabla g_i(w) - \nabla\lambda_i(w)$ for $i \in P'_=$.*
   - *Fill $\mathcal{A}_B$ from the remaining activities, until rank$[\mathcal{A}_B]$ = rank$[\mathcal{A}]$ and $\mathcal{A}_B$ has rank$[\mathcal{A}]$ columns.*
2. *Fill the corresponding nonbasic activity matrix $\mathcal{A}_N$ with the gradients of the nonbasic activities.*

3. *Solve for Ψ: $\mathcal{A}_B\Psi = \gamma$, where γ is the gradient of the differentiable part of p_μ at the current point.*

4. *Express the gradients of the nonbasic activities in terms of the gradients of the basic activities by solving for the matrix κ in the system of equations $\mathcal{A}_B\kappa = \mathcal{A}_N$.*

5. *Express first order change in the nonbasic activities in terms of the basic activities.*

6. *Express first order change in the penalty function in terms of the change in the basic activities using Ψ and κ.*

7. *Derive a set of necessary conditions on Ψ and κ to guarantee that the first order change in the penalty function is nonnegative along any direction d.*

These steps are now examined in greater detail.

6.6.1 Building the Basic and Nonbasic Matrices

Because the gradients $\nabla\lambda_i(w)$ are identity columns, the gradients of the activities listed in the first part of Step 1 above form a linearly independent set. Consequently, the submatrix of $\mathcal{A}$ consisting of these activities has full rank and this submatrix defines the initial columns of $\mathcal{A}_B$. The basis is filled out by examining the remaining gradients in the following order: $\nabla g_i(w)$ for $i \in P^0_=$, $\nabla g_i(w)$ for $i \in P^0_g$, $\nabla G_i(w)$ for $i \in T_0$, and $\nabla c_i(w)$ for $i \in M_0$. A candidate gradient a is added to the current basis matrix $\mathcal{A}_B$ if it is judged that the addition of the vector increases the rank of $\mathcal{A}_B$. If a_i lies in the range of the current basis matrix, then the current nonbasic matrix gets augmented with a. Otherwise, a is placed in the current basic matrix. Algorithm 6.3 details the process.

Algorithm 6.3 (Building the Basic Activity Matrix)

- *REPEAT*

  1. *Examine candidate gradient a, which is not currently in either $\mathcal{A}_B$ or $\mathcal{A}_N$.*
  2. *Solve, in a least squares sense, $\mathcal{A}_B\psi = a$, to determine the relationship between a and $\mathcal{A}_B$.*
  3. *Calculate the residual vector $r = \mathcal{A}_B\psi - a$.*
  4. *If $r = 0$, then $\mathcal{A}_N = [\mathcal{A}_N, a]$, else $\mathcal{A}_B = [\mathcal{A}_B, a]$.*

- *UNTIL $Rank[\mathcal{A}_B] = Rank[\mathcal{A}]$.*

In practice, the test on r in Step 4 above is typically relaxed to account for roundoff error.

The choice of $\mathcal{A}_B$ and $\mathcal{A}_N$ may not be unique. However, as long as they are defined using the process described in Algorithm 6.2, the conditions developed in the rest of the chapter are applicable.

Additional notation is required to specify the composition of the basic and nonbasic matrices.

Definition 6.1 *Given $\mathcal{A}_B$ and $\mathcal{A}_N$, the following sets are defined.*

- *Let $B(T_0)$ be the set of indices $i \in T_0$ for which $\nabla G_i(w)$ is in $\mathcal{A}_B$.*
- *Let $N(T_0) = T_0 \setminus B(T_0)$ be the indices of the nonbasic activities in T_0.*
- *Let $B(P_g^0)$ be the set of indices $i \in P_g^0$ for which $\nabla g_i(w)$ is in $\mathcal{A}_B$.*
- *Let $N(P_g^0) = P_g^0 \setminus B(P_g^0)$ be the indices of the nonbasic activities in P_g^0.*
- *Let $B(P_=^0)$ be the set of indices $i \in P_=^0$ for which $\nabla g_i(w)$ and $\nabla\lambda_i(w)$ are in $\mathcal{A}_B$.*

- *Let* $N(P^0_=) = P^0_= \setminus B(P^0_=)$ *be the indices* $i \in P^0_=$ *for which only* $\nabla\lambda_i(w)$ *is in* $\mathcal{A}_B$.
- *Let* $B(M_0)$ *be the set of indices* $i \in M_0$ *for which* $\nabla c_i(w)$ *is in* $\mathcal{A}_B$.
- *Let* $N(M_0) = M_0 \setminus B(M_0)$ *be the indices of the nonbasic activities in* M_0.

6.6.2 Solving for the Multipliers

Both sets of multipliers Ψ and κ can be efficiently computed using a single factorization of $\mathcal{A}_B$. Let Q be an orthogonal matrix and R be an upper triangular matrix which satisfy $\mathcal{A}_B = QR$. The vector Ψ is the solution of the triangular system $R\Psi = Q^T\gamma$. The matrix κ is the solution to the system $R\kappa = Q^T\mathcal{A}_N$, which is easily determined column by column by solving triangular system of equations. However, each column in κ can be computed in Step 2 of Algorithm 6.3 (when $r = 0$) by simply augmenting that solution with zeros.

The vector Ψ and the matrix κ are unique for any given choice of $\mathcal{A}_B$ and $\mathcal{A}_N$.

Definition 6.2 *The components of the matrix* κ *need to be accessed for basic and nonbasic activities.*

1. *For the nonbasic activities, let*
   - κ^G_i, *for* $i \in N(T^0)$, *refer to the column of* κ *corresponding to the multipliers relating the dependence of* $\nabla G_i(w)$ *on the basic activities,*
   - κ^g_i, *for* $i \in N(P^0_g) \cup N(P^0_=)$, *refer to the column of* κ *corresponding to the multipliers relating the dependence of* $\nabla g_i(w)$ *on the basic activities,*
   - κ^c_i, *for* $i \in N(M^0)$, *refer to the column of* κ *corresponding to the multipliers relating the dependence of* $\nabla c_i(w)$ *on the basic activities.*

2. For the basic activities, let

- $\kappa[G_j]$, *for* $j \in B(T^0)$, *refer to the row of* κ *corresponding to the multipliers relating the dependence of the nonbasic activities on* $\nabla G_j(w)$.
- $\kappa[\lambda_j]$, *for* $j \in P_\lambda^0 \cup P_=^0$, *refer to the row of* κ *corresponding to the multipliers relating the dependence of the nonbasic activities on* $\nabla \lambda_j(w)$.
- $\kappa[g_j - \lambda_j]$, *for* $j \in P_=^{\prime}$, *refer to the row of* κ *corresponding to the multipliers relating the dependence of the nonbasic activities on* $\nabla g_j(w) - \nabla \lambda_j(w)$.
- $\kappa[g_j]$, *for* $j \in B(P_g^0) \cup B(P_=^0)$, *refer to the row of* κ *corresponding to the multipliers relating the dependence of the nonbasic activities on* $\nabla g_j(w)$.
- $\kappa[c_j]$, *for* $j \in B(M^0)$, *refer to the row of* κ *corresponding to the multipliers relating the dependence of the nonbasic activities on* $\nabla c_j(w)$.

3. For a nonbasic activity n_i *and a basic activity* b_j, $\kappa_i^n[b_j]$ *is the multiplier relating the dependence of* n_i *on* b_j.

6.6.3 First Order Change in the Nonbasic Activities

The first order change in any nonbasic activity can be expressed in terms of the first order change in the basic activities. For $i \in N(T^0)$ and any direction d,

$$\begin{aligned}
d^T \nabla G_i(w) &= d^T A_B \kappa_i^G \\
&= \sum_{B(T^0)} \kappa_i^G[G_j] d^T \nabla G_j(w) + \sum_{P_\lambda^0} \kappa_i^G[\lambda_j] d^T \nabla \lambda_j(w) \\
&\quad + \sum_{P_=^{\prime}} \kappa_i^G[g_j - \lambda_j] d^T (\nabla g_j(w) - \nabla \lambda_j(w)) + \sum_{B(P_g^0)} \kappa_i^G[g_j] d^T \nabla g_j(w) \\
&\quad + \sum_{P_=^0} \kappa_i^G[\lambda_j] d^T \nabla \lambda_j(w) + \sum_{B(P_=^0)} \kappa_i^G[g_j] d^T \nabla g_j(w) \\
&\quad + \sum_{B(M^0)} \kappa_i^G[c_j] d^T \nabla c_j(w),
\end{aligned}$$

where each of the summations is over the index j.

First order changes in the remaining nonbasic activities can be similarly expressed using κ.

6.6.4 First Order Change in the Penalty Function

Recall from Lemma 3.6 that, for any direction d and step $0 \leq \alpha < \alpha_1$, where α_1 is the first breakpoint of an inactivity along d,

$$p_\mu(w + \alpha d) = \delta(w + \alpha d) + \mu\eta(w + \alpha d)$$

where δ is the differentiable part of p_μ at w (which is differentiable over $\alpha \in [0, \alpha_1)$) and η is the corresponding nondifferentiable part of p_μ. At a degenerate stationary point w, we can write, using the measure of curvature introduced in Section 3.5.2,

$$\begin{aligned}
\delta(w+\alpha d) &= p_\mu(w) + \alpha d^T \gamma + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega(\delta, w, \alpha d) \\
&= p_\mu(w) + \alpha d^T A_B \Psi + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega(\delta, w, \alpha d), \\
\eta(w+\alpha d) &= -\sum_{T^0} \min(G_i(w+\alpha d), 0) + \sum_{P_\lambda^0} |\lambda_i(w+\alpha d)| + \sum_{P_g^0} |g_i(w+\alpha d)| \\
&\quad + \sum_{P'_=} \text{sign}[\lambda_i(w)] \min(0, g_i(w+\alpha d) - \lambda_i(w+\alpha d)) \\
&\quad + \sum_{P_=^0} |\min(\lambda_i(w+\alpha d), g_i(w+\alpha d))| + \sum_{M^0} |c_i(w+\alpha d)| \\
&= -\alpha \sum_{T^0} \min(d^T \nabla G_i(w), 0) + \alpha \sum_{P_\lambda^0} |d^T \nabla \lambda_i(w)| + \alpha \sum_{P_g^0} |d^T \nabla g_i(w)| \\
&\quad + \alpha \sum_{P'_=} \text{sign}[\lambda_i(w)] \min(0, d^T(\nabla g_i(w) - \nabla \lambda_i(w))) \\
&\quad + \alpha \sum_{P_=^0} |\min(d^T \nabla \lambda_i(w), d^T \nabla g_i(w))| + \alpha \sum_{M^0} |d^T \nabla c_i(w)| \\
&\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_\eta(w, \alpha d) \\
&= -\alpha \sum_{B(T^0)} \min(d^T \nabla G_i(w), 0) - \alpha \sum_{N(T^0)} \min(d^T A_B \kappa_i^G, 0)
\end{aligned}$$

$$
\begin{aligned}
&+\alpha \sum_{P_\lambda^0} |d^T \nabla \lambda_i(w)| + \alpha \sum_{B(P_g^0)} |d^T \nabla g_i(w)| + \alpha \sum_{N(P_g^0)} |d^T A_B \kappa_i^g| \\
&+\alpha \sum_{P'_=} \text{sign}[\lambda_i(w)] \min(0, d^T(\nabla g_i(w) - \nabla \lambda_i(w))) \\
&+\alpha \sum_{B(P_=^0)} |\min(d^T \nabla \lambda_i(w), d^T \nabla g_i(w))| \\
&+\alpha \sum_{N(P_=^0)} |\min(d^T \nabla \lambda_i(w), d^T A_B \kappa_i^g)| \\
&+\alpha \sum_{B(M^0)} |d^T \nabla c_i(w)| + \alpha \sum_{N(M^0)} |d^T A_B \kappa_i^c| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_\eta(w, \alpha d),
\end{aligned}
$$

where the individual summations are over index i.

So, for $0 \leq \alpha < \alpha_1$,

$$
\begin{aligned}
p_\mu(w + \alpha d) \;=\; & p_\mu(w) + \alpha d^T A_B \Psi - \alpha\mu \sum_{B(T^0)} \min(d^T \nabla G_i(w), 0) \\
&-\alpha\mu \sum_{N(T^0)} \min(d^T A_B \kappa_i^G, 0) + \alpha\mu \sum_{P_\lambda^0} |d^T \nabla \lambda_i(w)| \\
&+\alpha\mu \sum_{B(P_g^0)} |d^T \nabla g_i(w)| + \alpha\mu \sum_{N(P_g^0)} |d^T A_B \kappa_i^g| \\
&+\alpha\mu \sum_{P'_=} \text{sign}[\lambda_i(w)] \min(0, d^T(\nabla g_i(w) - \nabla \lambda_i(w))) \\
&+\alpha\mu \sum_{B(P_=^0)} |\min(d^T \nabla \lambda_i(w), d^T \nabla g_i(w))| \\
&+\alpha\mu \sum_{N(P_=^0)} |\min(d^T \nabla \lambda_i(w), d^T A_B \kappa_i^g)| + \alpha\mu \sum_{B(M^0)} |d^T \nabla c_i(w)| \\
&+\alpha\mu \sum_{N(M^0)} |d^T A_B \kappa_i^c| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \qquad (6.1)
\end{aligned}
$$

where $\Omega_{p_\mu}(w, \alpha d) = \Omega(\delta, w, \alpha d) + \mu \Omega_\eta(w, \alpha d)$.

6.6.5 Deriving First Order Optimality Conditions

A set of first order optimality conditions which must be satisfied at a degenerate minimum of p_μ are derived using the above expression for $p_\mu(w + \alpha d)$, where $0 \leq \alpha < \alpha_1$, and are presented in three separate groupings.

Definition 6.3 *A direction d is a first order descent direction for the penalty function if the first order change in the differentiable part of p_μ along d from the current point has a negative value.*

Group One

In this section, a set of necessary optimality conditions at a degenerate first order point of the penalty function are obtained by analyzing the effect on p_μ of dropping a single basic activity.

Definition 6.4

- *For $j \in B(T^0)$, let*

$$\begin{aligned}
\zeta_j^G &= \sum_{N(P_g^0)} |\kappa_i^g[G_j]| + \sum_{N(M^0)} |\kappa_i^c[G_j]| \\
u_j^G &= \sum_{N(T^0)} \max(\kappa_i^G[G_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[G_j], 0) + \zeta_j^G \\
v_j^G &= -\sum_{N(T^0)} \min(\kappa_i^G[G_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[G_j], 0) + \zeta_j^G.
\end{aligned}$$

- *For $j \in P'_= = P_=^+ \cup P_=^-$, let*

$$\begin{aligned}
\zeta_j^{g-\lambda} &= \sum_{N(P_g^0)} |\kappa_i^g[g_j - \lambda_j]| + \sum_{N(M^0)} |\kappa_i^c[g_j - \lambda_j]| \\
u_j^{g-\lambda} &= \sum_{N(T^0)} \max(\kappa_i^G[g_j - \lambda_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[g_j - \lambda_j], 0) + \zeta_j^{g-\lambda} \\
v_j^{g-\lambda} &= -\sum_{N(T^0)} \min(\kappa_i^G[g_j - \lambda_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[g_j - \lambda_j], 0) + \zeta_j^{g-\lambda}
\end{aligned}$$

- *For $j \in B(P_=^0)$, let*

$$\zeta_j^\lambda = \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| + \sum_{N(M^0)} |\kappa_i^c[\lambda_j]|$$

$$\begin{aligned}
u_j^\lambda &= \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda \\
v_j^\lambda &= -\sum_{N(T^0)} \min(\kappa_i^G[\lambda_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda \\
\zeta_j^g &= \sum_{N(P_g^0)} |\kappa_i^g[g_j]| + \sum_{N(M^0)} |\kappa_i^c[g_j]| \\
u_j^g &= \sum_{N(T^0)} \max(\kappa_i^G[g_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[g_j], 0) + \zeta_j^g \\
v_j^g &= -\sum_{N(T^0)} \min(\kappa_i^G[g_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[g_j], 0) + \zeta_j^g
\end{aligned}$$

- *For $j \in P_\lambda^0$, let*

$$\begin{aligned}
\zeta_j^\lambda &= \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| + \sum_{N(M^0)} |\kappa_i^c[\lambda_j]| \\
u_j^\lambda &= \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda \\
v_j^\lambda &= -\sum_{N(T^0)} \min(\kappa_i^G[\lambda_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda
\end{aligned}$$

- *For $j \in B(P_g^0)$, let*

$$\begin{aligned}
\zeta_j^g &= \sum_{N(P_g^0)} |\kappa_i^g[g_j]| + \sum_{N(M^0)} |\kappa_i^c[g_j]| \\
u_j^g &= \sum_{N(T^0)} \max(\kappa_i^G[g_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[g_j], 0) + \zeta_j^g \\
v_j^g &= -\sum_{N(T^0)} \min(\kappa_i^G[g_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[g_j], 0) + \zeta_j^g
\end{aligned}$$

- *For $j \in B(M^0)$, let*

$$\begin{aligned}
\zeta_j^c &= \sum_{N(P_g^0)} |\kappa_i^g[c_j]| + \sum_{N(M^0)} |\kappa_i^c[c_j]| \\
u_j^c &= \sum_{N(T^0)} \max(\kappa_i^G[c_j], 0) + \sum_{N(P_=^0)} \max(\kappa_i^g[c_j], 0) + \zeta_j^c \\
v_j^c &= -\sum_{N(T^0)} \min(\kappa_i^G[c_j], 0) - \sum_{N(P_=^0)} \min(\kappa_i^g[c_j], 0) + \zeta_j^c
\end{aligned}$$

- *For $j \in N(P^0_=)$, let*

$$
\begin{aligned}
\zeta_j^\lambda &= \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| + \sum_{N(M^0)} |\kappa_i^c[\lambda_j]| \\
u_j^\lambda &= \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j], 0) + \sum_{N(P^0_=)\backslash\{j\}} \max(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda \\
v_j^\lambda &= -\sum_{N(T^0)} \min(\kappa_i^G[\lambda_j], 0) - \sum_{N(P^0_=)\backslash\{j\}} \min(\kappa_i^g[\lambda_j], 0) + \zeta_j^\lambda.
\end{aligned}
$$

Lemma 6.1 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $A_B^T d = \sigma e_j^G$, for some $j \in B(T^0)$, is a first order descent direction for p_μ from w if and only if*

$$
\begin{aligned}
&(i) \quad \sigma < 0 \quad \text{and} \quad \Psi_j^G - \mu u_j^G > \mu \\
\text{or} \quad &(ii) \quad \sigma > 0 \quad \text{and} \quad \Psi_j^G + \mu v_j^G < 0.
\end{aligned}
$$

Proof: For $0 \le \alpha < \alpha_1$, from equation (6.1),

$$
\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\sigma(e_j^G)^T\Psi - \alpha\mu\min(\sigma, 0) - \alpha\mu \sum_{N(T^0)} \min(\sigma(e_j^G)^T\kappa_i^G, 0) \\
&\quad + \alpha\mu \sum_{N(P_g^0)} |\sigma(e_j^G)^T\kappa_i^g| + \alpha\mu \sum_{N(P^0_=)} |\min(0, \sigma(e_j^G)^T\kappa_i^g)| \\
&\quad + \alpha\mu \sum_{N(M^0)} |\sigma(e_j^G)^T\kappa_i^c| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha\sigma\Psi_j^G - \alpha\mu\min(\sigma, 0) - \alpha\mu \sum_{N(T^0)} \min(\sigma\kappa_i^G[G_j], 0) \\
&\quad + \alpha\mu \sum_{N(P_g^0)} |\sigma||\kappa_i^g[G_j]| - \alpha\mu \sum_{N(P^0_=)} \min(\sigma\kappa_i^g[G_j], 0) \\
&\quad + \alpha\mu \sum_{N(M^0)} |\sigma||\kappa_i^c[G_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Without loss of generality, assume that $\sigma = \pm 1$. Therefore, consider the two cases separately.

1. $\sigma = -1$:

$$
p_\mu(w + \alpha d) = p_\mu(w) - \alpha\Psi_j^G + \alpha\mu + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[G_j], 0)
$$

$$
\begin{aligned}
&+\alpha\mu \sum_{N(P_{g}^{0})} |\kappa_i^g[G_j]| + \alpha\mu \sum_{N(P_{=}^{0})} \max(\kappa_i^g[G_j], 0) \\
&+\alpha\mu \sum_{N(M^0)} |\kappa_i^c[G_j]| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\
= \;& p_\mu(w) - \alpha\Psi_j^G + \alpha\mu + \alpha\mu u_j^G + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
= \;& p_\mu(w) - \alpha(\Psi_j^G - \mu u_j^G - \mu) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Therefore, first order descent is possible if and only if $\Psi_j^G - \mu u_j^G > \mu$.

2. $\sigma = +1$:

$$
\begin{aligned}
p_\mu(w + \alpha d) = \;& p_\mu(w) + \alpha\Psi_j^G - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[G_j], 0) \\
&+\alpha\mu \sum_{N(P_{g}^{0})} |\kappa_i^g[G_j]| - \alpha\mu \sum_{N(P_{=}^{0})} \min(\kappa_i^g[G_j], 0) \\
&+\alpha\mu \sum_{N(M^0)} |\kappa_i^c[G_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
= \;& p_\mu(w) + \alpha(\Psi_j^G + \mu v_j^G) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Therefore, first order descent is possible if and only if $\Psi_j^G + \mu v_j^G < 0$. □

Lemma 6.2 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^{g-\lambda}$, for some $j \in P_{=}^{-}$, is a first order descent direction for p_μ from w if and only if*

$$
\begin{aligned}
&(i) \quad \sigma < 0 \quad \text{and} \quad \Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} > \mu \\
\text{or} \quad &(ii) \quad \sigma > 0 \quad \text{and} \quad \Psi_j^{g-\lambda} + \mu v_j^{g-\lambda} < 0.
\end{aligned}
$$

Proof: Analogous to the proof of Lemma 6.1. □

Lemma 6.3 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^{\lambda}$, for some $j \in B(P_{=}^{0})$, is a first order descent direction for p_μ from w if and* *only if*

$$\begin{array}{lllll} & (i) & \sigma < 0 & \text{and} & \Psi_j^\lambda - \mu u_j^\lambda > \mu \\ \text{or} & (ii) & \sigma > 0 & \text{and} & \Psi_j^\lambda + \mu v_j^\lambda < 0. \end{array}$$

Proof: Analogous to the proof of Lemma 6.1. □

Lemma 6.4 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^g$, for some $j \in B(P_=^0)$, is a first order descent direction for p_μ from w if and only if*

$$\begin{array}{lllll} & (i) & \sigma < 0 & \text{and} & \Psi_j^g - \mu u_j^g > \mu \\ \text{or} & (ii) & \sigma > 0 & \text{and} & \Psi_j^g + \mu v_j^g < 0. \end{array}$$

Proof: Analogous to the proof of Lemma 6.1. □

Lemma 6.5 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^\lambda$, for some $j \in P_\lambda^0$, is a first order descent direction for p_μ from w if and only if*

$$\begin{array}{lllll} & (i) & \sigma < 0 & \text{and} & \Psi_j^\lambda - \mu u_j^\lambda > \mu \\ \text{or} & (ii) & \sigma > 0 & \text{and} & \Psi_j^\lambda + \mu v_j^\lambda < -\mu. \end{array}$$

Proof: For $0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\sigma\Psi_j^\lambda - \alpha\mu \sum_{N(T^0)} \min(\sigma\kappa_i^G[\lambda_j], 0) + \alpha\mu|\sigma| \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\sigma||\kappa_i^g[\lambda_j]| - \alpha\mu \sum_{N(P_=^0)} \min(0, \sigma\kappa_i^g[\lambda_j]) \\ &\quad + \alpha\mu \sum_{N(M^0)} |\sigma||\kappa_i^c[\lambda_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Without loss of generality, assume that $\sigma = \pm 1$.

1. $\sigma = -1$:

$$\begin{aligned} p_\mu(w+\alpha d) &= p_\mu(w) - \alpha\Psi_j^\lambda + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j], 0) + \alpha\mu \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| + \alpha\mu \sum_{N(P_{=}^0)} \max(\kappa_i^g[\lambda_j], 0) \\ &\quad + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j]| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\ &= p_\mu(w) - \alpha\Psi_j^\lambda + \alpha\mu u_j^\lambda + \alpha\mu + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\ &= p_\mu(w) - \alpha(\Psi_j^\lambda - \mu u_j^\lambda - \mu) + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Therefore, first order descent is possible if and only if $\Psi_j^\lambda - \mu u_j^\lambda > \mu$.

2. $\sigma = +1$:

$$\begin{aligned} p_\mu(w+\alpha d) &= p_\mu(w) + \alpha\Psi_j^\lambda - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[\lambda_j], 0) + \alpha\mu \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| - \alpha\mu \sum_{N(P_{=}^0)} \min(\kappa_i^g[\lambda_j], 0) \\ &\quad + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j]| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\ &= p_\mu(w) + \alpha(\Psi_j^\lambda + \mu v_j^\lambda + \mu) + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Therefore, first order descent is possible if and only if $\Psi_j^\lambda + \mu v_j^\lambda < -\mu$. □

Lemma 6.6 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $A_B^T d = \sigma e_j^g$, for some $j \in B(P_g^0)$, is a first order descent direction for p_μ from w if and only if*

$$\begin{aligned} &(i) \quad \sigma < 0 \quad \text{and} \quad \Psi_j^g - \mu u_j^g > \mu \\ \text{or} \quad &(ii) \quad \sigma > 0 \quad \text{and} \quad \Psi_j^g + \mu v_j^g < -\mu. \end{aligned}$$

Proof: Analogous to the proof of Lemma 6.5. □

Lemma 6.7 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^c$, for some $j \in B(M^0)$, is a first order descent direction for p_μ from w if and only if*

$$\begin{aligned} &(i) \quad \sigma < 0 \quad \text{and} \quad \Psi_j^c - \mu u_j^c > \mu \\ \text{or} \quad &(ii) \quad \sigma > 0 \quad \text{and} \quad \Psi_j^c + \mu v_j^c < -\mu. \end{aligned}$$

Proof: Analogous to the proof of Lemma 6.5. □

Lemma 6.8 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^{g-\lambda}$, for some $j \in P_=^+$, is a first order descent direction for p_μ from w if and only if*

$$\begin{aligned} &(i) \quad \sigma < 0 \quad \text{and} \quad \Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} > -\mu \\ \text{or} \quad &(ii) \quad \sigma > 0 \quad \text{and} \quad \Psi_j^{g-\lambda} + \mu v_j^{g-\lambda} < 0. \end{aligned}$$

Proof: For $0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\sigma\Psi_j^{g-\lambda} - \alpha\mu \sum_{N(T^0)} \min(\sigma\kappa_i^G[g_j - \lambda_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_s^0)} |\sigma||\kappa_i^g[g_j - \lambda_j]| + \alpha\mu \min(0, \sigma) \\ &\quad - \alpha\mu \sum_{N(P_=^0)} \min(0, \sigma\kappa_i^g[g_j - \lambda_j]) + \alpha\mu \sum_{N(M^0)} |\sigma||\kappa_i^c[g_j - \lambda_j]| \\ &\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Without loss of generality, assume that $\sigma = \pm 1$.

1. $\sigma = -1$:

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) - \alpha\Psi_j^{g-\lambda} + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[g_j - \lambda_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_s^0)} |\kappa_i^g[g_j - \lambda_j]| - \alpha\mu + \alpha\mu \sum_{N(P_=^0)} \max(\kappa_i^g[g_j - \lambda_j], 0) \end{aligned}$$

$$
\begin{aligned}
&+\alpha\mu \sum_{N(M^0)} |\kappa_i^c[g_j - \lambda_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) - \alpha(\Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} + \mu) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Therefore, first order descent is possible only if and only if $\Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} > -\mu$.

2. $\sigma = +1$:

$$
\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\Psi_j^{g-\lambda} - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[g_j - \lambda_j], 0) \\
&\quad +\alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[g_j - \lambda_j]| - \alpha\mu \sum_{N(P_=^0)} \min(\kappa_i^g[g_j - \lambda_j], 0) \\
&\quad +\alpha\mu \sum_{N(M^0)} |\kappa_i^c[g_j - \lambda_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi_j^{g-\lambda} + \mu v_j^{g-\lambda}) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Therefore, first order descent is possible only if and only if $\Psi_j^{g-\lambda} + \mu v_j^{g-\lambda} < 0$. □

Lemma 6.9 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma e_j^\lambda$, for some $j \in N(P_=^0)$, is a first order descent direction for p_μ from w if and only if*

$$
\begin{aligned}
&\textit{(i)} \quad \sigma < 0 \quad \textit{and} \quad \Psi_j^\lambda - \mu u_j^\lambda - \mu \max(1, w_j^g[\lambda_j]) > 0 \\
\textit{or} \quad &\textit{(ii)} \quad \sigma > 0 \quad \textit{and} \quad \Psi_j^\lambda + \mu v_j^\lambda + \mu|\min(1, w_j^g[\lambda_j])| < 0.
\end{aligned}
$$

Proof: For $0 \le \alpha < \alpha_1$,

$$
\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\sigma\Psi_j^\lambda - \alpha\mu \sum_{N(T^0)} \min(\sigma\kappa_i^G[\lambda_j], 0) \\
&\quad +\alpha\mu \sum_{N(P_g^0)} |\sigma||\kappa_i^g[\lambda_j]| - \alpha\mu \sum_{N(P_=^0)\setminus\{j\}} \min(0, \sigma\kappa_i^G[\lambda_j]) \\
&\quad +\alpha\mu|\min(\sigma, \sigma\kappa_j^g[\lambda_j])| + \alpha\mu \sum_{N(M^0)} |\sigma||\kappa_i^g[\lambda_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

Without loss of generality, assume that $\sigma = \pm 1$.

1. $\sigma = -1$:

$$\begin{aligned}
p_\mu(w+\alpha d) &= p_\mu(w) - \alpha\Psi_j^\lambda + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j], 0) \\
&\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| + \alpha\mu \sum_{N(P_=^0)\setminus\{j\}} \max(0, \kappa_i^G[\lambda_j]) \\
&\quad + \alpha\mu|\min(-1, -\kappa_j^g[\lambda_j])| + \alpha\mu \sum_{N(M^0)} |\kappa_i^g[\lambda_j]| \\
&\quad + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) - \alpha\Psi_j^\lambda + \alpha\mu u_j^\lambda + \alpha\mu| - \max(1, \kappa_j^g[\lambda_j])| \\
&\quad + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) - \alpha(\Psi_j^\lambda - \mu u_j^\lambda - \mu\max(1, \kappa_j^g[\lambda_j])) \\
&\quad + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}$$

Therefore, first order descent is possible if and only if

$$\Psi_j^\lambda - \mu u_j^\lambda - \mu\max(1, \kappa_j^g[\lambda_j]) > 0.$$

2. $\sigma = +1$:

$$\begin{aligned}
p_\mu(w+\alpha d) &= p_\mu(w) + \alpha\Psi_j^\lambda - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[\lambda_j], 0) \\
&\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j]| - \alpha\mu \sum_{N(P_=^0)\setminus\{j\}} \min(0, \kappa_i^G[\lambda_j]) \\
&\quad + \alpha\mu|\min(1, \kappa_j^g[\lambda_j])| + \alpha\mu \sum_{N(M^0)} |\kappa_i^g[\lambda_j]| \\
&\quad + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi_j^\lambda + \mu v_j^\lambda + \mu|\min(1, \kappa_j^g[\lambda_j])|) \\
&\quad + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}$$

Therefore, first order descent is possible if and only if

$$\Psi_j^\lambda + \mu v_j^\lambda + \mu|\min(1, \kappa_j^g[\lambda_j])| < 0. \qquad \square$$

Group Two

In this section, a set of necessary optimality conditions are derived by analyzing the effect on the penalty function p_μ of dropping both $\lambda_j(w) = 0$ and $g_j(w) = 0$, for some $j \in B(P^0_=)$, when $|d^T \nabla \lambda_j(w)| = |d^T \nabla g_j(w)|$.

The following definitions involving sums of generalized multipliers are required.

Definition 6.5 *For* $j \in B(P^0_=)$,

$$\begin{aligned}
\zeta_j^{\lambda+g} &= \sum_{N(P^0_g)} |\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \sum_{N(M^0)} |\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]| \\
u_j^{\lambda+g} &= \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) + \sum_{N(P^0_=)} \max(\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) + \zeta_j^{\lambda+g} \\
v_j^{\lambda+g} &= -\sum_{N(T^0)} \min(\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) - \sum_{N(P^0_=)} \min(\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) + \zeta_j^{\lambda+g} \\
\zeta_j^{\lambda-g} &= \sum_{N(P^0_g)} |\kappa_i^g[\lambda_j] - \kappa_i^g[g_j]| + \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \kappa_i^c[g_j]| \\
u_j^{\lambda-g} &= \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j] - \kappa_i^G[g_j], 0) + \sum_{N(P^0_=)} \max(\kappa_i^g[\lambda_j] - \kappa^g[g_j], 0) + \zeta_j^{\lambda-g} \\
v_j^{\lambda-g} &= -\sum_{N(T^0)} \min(\kappa_i^G[\lambda_j] - \kappa_i^G[g_j], 0) - \sum_{N(P^0_=)} \min(\kappa_i^g[\lambda_j] - \kappa_i^g[g_j], 0) + \zeta_j^{\lambda-g}.
\end{aligned}$$

Lemma 6.10 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $\mathcal{A}_B^T d = \sigma(e_j^\lambda + e_j^g)$, for some $j \in B(P^0_=)$, is a first order descent direction for p_μ from w if and only if*

$$\begin{aligned}
&(i) \quad \sigma < 0 \quad \textit{and} \quad \Psi_j^\lambda + \Psi_j^g - \mu u_j^{\lambda+g} > \mu \\
\textit{or} \quad &(ii) \quad \sigma > 0 \quad \textit{and} \quad \Psi_j^\lambda + \Psi_j^g + \mu v_j^{\lambda+g} < -\mu.
\end{aligned}$$

Proof: For $0 \le \alpha < \alpha_1$,

$$\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha\sigma(\Psi_j^\lambda + \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\sigma(\kappa_i^G[\lambda_j] + \kappa_i^G[g_j]), 0) \\
&\quad + \alpha\mu \sum_{N(P^0_g)} |\sigma||\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \alpha\mu|\sigma|
\end{aligned}$$

$$-\alpha\mu \sum_{N(P^0_=)} \min(\sigma(\kappa^g_i[\lambda_j] + \kappa^g_i[g_j]), 0)$$
$$+\alpha\mu \sum_{N(M^0)} |\sigma||\kappa^c_i[\lambda_j] + \kappa^c_i[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).$$

Without loss of generality, assume that $\sigma = \pm 1$.

1. $\sigma = -1$:

$$\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) - \alpha(\Psi^\lambda_j + \Psi^g_j) + \alpha\mu \sum_{N(T^0)} \max(\kappa^G_i[\lambda_j] + \kappa^G_i[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(P^0_s)} |\kappa^g_i[\lambda_j] + \kappa^g_i[g_j]| + \alpha\mu \\
&\quad + \alpha\mu \sum_{N(P^0_=)} \max(\kappa^g_i[\lambda_j] + \kappa^g_i[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(M^0)} |\kappa^c_i[\lambda_j] + \kappa^c_i[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) - \alpha(\Psi^\lambda_j + \Psi^g_j) + \alpha\mu u_j^{\lambda+g} + \alpha\mu + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) - \alpha(\Psi^\lambda_j + \Psi^g_j - \mu u_j^{\lambda+g} - \mu) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}$$

   Therefore, first order descent is possible if and only if

$$\Psi^\lambda_j + \Psi^g_j - \mu u_j^{\lambda+g} > \mu.$$

2. $\sigma = +1$:

$$\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha(\Psi^\lambda_j + \Psi^g_j) - \alpha\mu \sum_{N(T^0)} \min(\kappa^G_i[\lambda_j] + \kappa^G_i[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(P^0_s)} |\kappa^g_i[\lambda_j] + \kappa^g_i[g_j]| + \alpha\mu \\
&\quad - \alpha\mu \sum_{N(P^0_=)} \min(\kappa^g_i[\lambda_j] + \kappa^g_i[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(M^0)} |\kappa^c_i[\lambda_j] + \kappa^c_i[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi^\lambda_j + \Psi^g_j) + \alpha\mu v_j^{\lambda+g} + \alpha\mu + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi^\lambda_j + \Psi^g_j + \mu v_j^{\lambda+g} + \mu) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}$$

Therefore, first order descent is possible if and only if

$$\Psi_j^\lambda + \Psi_j^g + \mu v_j^{\lambda+g} < -\mu. \qquad \square$$

Lemma 6.11 *If w is a stationary, degenerate point of p_μ, then any direction d satisfying $A_B^T d = \sigma(e_j^\lambda - e_j^g)$, for some $j \in B(P_=^0)$, is a first order descent direction for p_μ from w if and only if*

$$\begin{aligned} &(i) \quad \sigma < 0 \quad \textit{and} \quad \Psi_j^\lambda - \Psi_j^g - \mu u_j^{\lambda-g} > \mu \\ \textit{or} \quad &(ii) \quad \sigma > 0 \quad \textit{and} \quad \Psi_j^\lambda - \Psi_j^g + \mu v_j^{\lambda-g} < -\mu. \end{aligned}$$

Proof: For $0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w+\alpha d) &= p_\mu(w) + \alpha\sigma(\Psi_j^\lambda - \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\sigma(\kappa_i^G[\lambda_j] - \kappa_i^G[g_j]), 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\sigma||\kappa_i^g[\lambda_j] - \kappa_i^g[g_j]| + \alpha\mu|\sigma| \\ &\quad - \alpha\mu \sum_{N(P_=^0)} \min(\sigma(\kappa_i^g[\lambda_j] - \kappa_i^g[g_j]), 0) \\ &\quad + \alpha\mu \sum_{N(M^0)} |\sigma||\kappa_i^c[\lambda_j] - \kappa_i^c[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Without loss of generality, assume that $\sigma = \pm 1$.

1. $\sigma = -1$:

$$\begin{aligned} p_\mu(w+\alpha d) &= p_\mu(w) - \alpha(\Psi_j^\lambda - \Psi_j^g) + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j] - \kappa_i^G[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j] - \kappa_i^g[g_j]| + \alpha\mu \\ &\quad + \alpha\mu \sum_{N(P_=^0)} \max(\kappa_i^g[\lambda_j] - \kappa_i^g[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \kappa_i^c[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\ &= p_\mu(w) - \alpha(\Psi_j^\lambda - \Psi_j^g) + \alpha\mu u_j^{\lambda-g} + \alpha\mu + \frac{1}{2}\alpha^2\|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\ &= p_\mu(w) - \alpha(\Psi_j^\lambda - \Psi_j^g - \mu u_j^{\lambda-g} - \mu) + \frac{1}{2}\alpha^2\|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

Therefore, first order descent is possible if and only if

$$\Psi_j^\lambda - \Psi_j^g - \mu u_j^{\lambda-g} > \mu.$$

2. $\sigma = +1$:

$$\begin{aligned}
p_\mu(w+\alpha d) &= p_\mu(w) + \alpha(\Psi_j^\lambda - \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[\lambda_j] - \kappa_i^G[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j] - \kappa_i^g[g_j]| + \alpha\mu \\
&\quad - \alpha\mu \sum_{N(P_=^0)} \min(\kappa_i^g[\lambda_j] - \kappa_i^g[g_j], 0) \\
&\quad + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \kappa_i^c[g_j]| + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi_j^\lambda - \Psi_j^g) + \alpha\mu v_j^{\lambda-g} + \alpha\mu + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d) \\
&= p_\mu(w) + \alpha(\Psi_j^\lambda - \Psi_j^g + \mu v_j^{\lambda-g} + \mu) + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d).
\end{aligned}$$

Therefore, first order descent is possible if and only if

$$\Psi_j^\lambda - \Psi_j^g + \mu v_j^{\lambda-g} < -\mu. \qquad \square$$

Group Three

In this section, consider the effect on the penalty function of dropping, for some $j \in B(P_=^0)$, both $\lambda_j(w) = 0$ and $g_j(w) = 0$ when $|d^T \nabla \lambda_j(w)| \neq |d^T \nabla g_j(w)|$. Through analysis, a set of necessary conditions at a degenerate minimum point of p_μ will be derived.

The following group of definitions are required. These definitions concern the shape of the penalty function along directions d as described above. The shape changes as the differentiability of the penalty terms changes.

Definition 6.6 *For $j \in B(P^0_=)$:*

1. *Define breakpoints $\tau^1[j]$ as follows:*

$$i \in N(T^0): \quad \tau^1_i[j] = \begin{cases} - \quad \kappa^G_i[\lambda_j]/\kappa^G_i[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P^0_g): \quad \tau^1_i[j] = \begin{cases} - \quad \kappa^g_i[\lambda_j]/\kappa^g_i[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P^0_=): \quad \tau^1_i[j] = \begin{cases} - \quad \kappa^g_i[\lambda_j]/\kappa^g_i[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau^1_i[j] = \begin{cases} - \quad \kappa^c_i[\lambda_j]/\kappa^c_i[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise.} \end{cases}$$

2. *Let k^1_j be the number of distinct breakpoints in the interval $(0,1)$, and let these distinct breakpoints, denoted $t^1_l[j]$ for $l = 1, \cdots, k^1_j$ be indexed to satisfy*

$$0 < t^1_1[j] < \cdots < t^1_{k^1_j}[j] < 1.$$

In addition, define the additional breakpoints $t^1_0[j] = 0$ and $t^1_{k^1_j+1}[j] = 1$.

3. *For each $l = 0 : k^1_j$, define $\xi^l_i[j]$ as follows:*

$$i \in N(T^0): \quad \xi^l_i[j] = \begin{cases} pos[\kappa^G_i[\lambda_j]] & \text{if } \tau^1_i[j] < t^1_l[j] \\ pos[\kappa^G_i[g_j]]. & \text{otherwise} \end{cases}$$

$$i \in N(P^0_g): \quad \xi^l_i[j] = \begin{cases} sign[\kappa^g_i[\lambda_j]] & \text{if } \tau^1_i[j] < t^1_l[j] \\ sign[\kappa^g_i[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P^0_=): \quad \xi^l_i[j] = \begin{cases} pos[\kappa^g_i[\lambda_j]] & \text{if } \tau^1_i[j] < t^1_l[j] \\ pos[\kappa^g_i[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \xi^l_i[j] = \begin{cases} sign[\kappa^c_i[\lambda_j]] & \text{if } \tau^1_i[j] < t^1_l[j] \\ sign[\kappa^c_i[g_j]] & \text{otherwise.} \end{cases}$$

4. *For each $l = 0 : k_j^1$, define quantities $s_\lambda^l[j]$ and $s_g^l[j]$ as follows:*

$$
\begin{aligned}
s_\lambda^l[j] &= \Psi_j^\lambda - \mu \sum_{N(T^0)} \xi_i^l[j]\kappa_i^G[\lambda_j] - \mu \sum_{N(P_g^0)} \xi_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_=^0)} \xi_i^l[j]\kappa_i^g[\lambda_j] \\
&\quad - \mu \sum_{N(M^0)} \xi_i^l[j]\kappa_i^c[\lambda_j] - \mu \\
s_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \xi_i^l[j]\kappa_i^G[g_j] - \mu \sum_{N(P_g^0)} \xi_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \xi_i^l[j]\kappa_i^g[g_j] \\
&\quad - \mu \sum_{N(M^0)} \xi_i^l[j]\kappa_i^c[g_j].
\end{aligned}
$$

These terms are used to express the first order change in the penalty function, in terms of the first order change in the basic and nonbasic activities.

5. *For each $l = 0 : k_j^1$, define the intervals $\mathcal{J}_l[j]$ and $\mathcal{J}_l^*[j]$ as follows:*

$$
\begin{aligned}
\mathcal{J}_l[j] &= \begin{cases} (-\infty, -s_\lambda^l[j]/s_g^l[j]) & \text{if } s_g^l[j] < 0 \\ (-s_\lambda^l[j]/s_g^l[j], \infty) & \text{if } s_g^l[j] > 0 \\ (-\infty, \infty) & \text{if } s_g^l[j] = 0 \text{ and } s_\lambda^l[j] > 0 \\ \emptyset & \text{otherwise,} \end{cases} \\
\mathcal{J}_l^*[j] &= (t_l^1[j], t_{l+1}^1[j]) \cap \mathcal{J}_l[j].
\end{aligned}
$$

The first interval gives the region in which first order decrease in attained for a term related to the penalty function. The second interval, which intersects the first region with the breakpoint interval in which the term is equivalent to p_μ, gives the range of values for which first order descent is actually possible in the penalty function.

Lemma 6.12 *If w is a degenerate, stationary point, then there exists a first-order descent direction d for p_μ from w satisfying $A_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < \sigma_g < 0$ if and only if there exists $l \in \{0 : k_j^1\}$ such that $\mathcal{J}_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $A_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < \sigma_g < 0$. Without loss of generality, assume that $\sigma_\lambda = -1$ and $\sigma_g = -\tau$ for some $\tau \in (0, 1)$. Then, for

$0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w+\alpha d) &= p_\mu(w) - \alpha(\Psi_j^\lambda + \tau\Psi_j^g) + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j] + \tau\kappa_i^G[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]| + \alpha\mu \\ &\quad + \alpha\mu \sum_{N(P_{=}^0)} \max(\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] + \tau\kappa_i^c[g_j]| \\ &\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

For any $\tau \in (t_l^1[j], t_{l+1}^1[j])$, $l = 0 : k_j^1$,

$$\begin{aligned} i \in N(T^0): \quad \text{pos}[\kappa_i^G[\lambda_j] + \tau\kappa_i^G[g_j]] &= \xi_i^l[j] \\ i \in N(P_g^0): \quad \text{sign}[\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]] &= \xi_i^l[j] \\ i \in N(P_{=}^0): \quad \text{pos}[\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]] &= \xi_i^l[j] \\ i \in N(M^0): \quad \text{sign}[\kappa_i^c[\lambda_j] + \tau\kappa_i^c[g_j]] &= \xi_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w+\alpha d) = p_\mu(w) - \alpha(s_\lambda^l[j] + \tau s_g^l[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{J}_l[j]$, then the coefficient of $(-\alpha)$ is positive. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^1[j], t_{l+1}^1[j])$.

Therefore, if $\mathcal{J}_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_\lambda < \sigma_g < 0$. However, if there exists some $l \in \{0 : k_j^1\}$ such that $\mathcal{J}_l^*[j] \ne \emptyset$, then for any $\tau \in \mathcal{J}_l^*$, a direction d satisfying $\mathcal{A}_B^T d = -(e_j^\lambda + \tau e_j^g)$ provides first order descent in p_μ from w. □

The meanings of the terms defined below, and in all similar definitions, are analogous to the meanings of the related terms in Definition 6.6.

Definition 6.7 *For $j \in B(P_{=}^0)$:*

1. *Define breakpoints $\tau^2[j]$ as follows:*

$$i \in N(T^0): \quad \tau_i^2[j] = \begin{cases} \kappa_i^G[g_j]/\kappa_i^G[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^2[j] = \begin{cases} \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \tau_i^2[j] = \begin{cases} \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^2[j] = \begin{cases} \kappa_i^c[g_j]/\kappa_i^c[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise.} \end{cases}$$

2. *Let* k_j^2 *be the number of distinct breakpoints in the interval* $(0,1)$, *and let these distinct breakpoints, denoted* $t_l^2[j]$ *for* $l = 1, \cdots, k_j^2$, *be indexed to satisfy*

$$0 < t_1^2[j] < \cdots < t_{k_j^2}^2[j] < 1.$$

In addition, define $t_0^2[j] = 0$ *and* $t_{k_j^2+1}^2[j] = 0$.

3. *For each* $l = 0 : k_j^2$, *define* $\pi^l[j]$ *as follows:*

$$i \in N(T^0): \quad \pi_i^l[j] = \begin{cases} neg[\kappa_i^G[g_j]] & \textit{if } \tau_i^2[j] < t_l^2[j] \\ pos[\kappa_i^G[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \pi_i^l[j] = \begin{cases} sign[\kappa_i^g[g_j]] & \textit{if } \tau_i^2[j] < t_l^2[j] \\ -sign[\kappa_i^g[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \pi_i^l[j] = \begin{cases} neg[\kappa_i^g[g_j]] & \textit{if } \tau_i^2[j] < t_l^2[j] \\ pos[\kappa_i^g[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \pi_i^l[j] = \begin{cases} sign[\kappa_i^c[g_j]] & \textit{if } \tau_i^2[j] < t_l^2[j] \\ -sign[\kappa_i^c[\lambda_j]] & \textit{otherwise.} \end{cases}$$

4. *For each* $l = 0 : k_j^2$, *define the quantities* $r_\lambda^l[j]$ *and* $r_g^l[j]$ *as follows:*

$$r_\lambda^l[j] = \Psi_j^\lambda - \mu \sum_{N(T^0)} \pi_i^l[j]\kappa_i^G[\lambda_j] + \mu \sum_{N(P_g^0)} \pi_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_=^0)} \pi_i^l[j]\kappa_i^g[\lambda_j]$$

$$
\begin{aligned}
& +\mu \sum_{N(M^0)} \pi_i^l[j]\kappa_i^c[\lambda_j] - \mu \\
r_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \pi_i^l[j]\kappa_i^G[g_j] + \mu \sum_{N(P_g^0)} \pi_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \pi_i^l[j]\kappa_i^g[g_j] \\
& +\mu \sum_{N(M^0)} \pi_i^l[j]\kappa_i^c[g_j].
\end{aligned}
$$

5. *For each $l = 0 : k_j^2$, define the intervals $\mathcal{K}_l[j]$ and $\mathcal{K}_l^*[j]$ as follows:*

$$
\begin{aligned}
\mathcal{K}_l[j] &= \begin{cases} (-\infty, r_g^l[j]/r_\lambda^l[j]) & \text{if } r_\lambda^l[j] < 0 \\ (r_g^l[j]/r_\lambda^l[j], \infty) & \text{if } r_\lambda^l[j] > 0 \\ (-\infty, \infty) & \text{if } r_\lambda^l[j] = 0 \text{ and } r_g^l[j] < 0 \\ \emptyset & \text{otherwise,} \end{cases} \\
\mathcal{K}_l^*[j] &= (t_l^2[j], t_{l+1}^2[j]) \cap \mathcal{K}_l[j].
\end{aligned}
$$

Lemma 6.13 *If w is a degenerate, stationary point, then there exists a first order descent direction d for p_μ from w satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < 0 < \sigma_g$ and $|\sigma_\lambda| < |\sigma_g|$ if and only if there exists $l \in \{0 : k_j^2\}$ such that $\mathcal{K}_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < 0 < \sigma_g$ and $|\sigma_\lambda| < |\sigma_g|$. Without loss of generality, assume that $\sigma_g = 1$ and $\sigma_\lambda = -\tau$ for some $\tau \in (0,1)$. Then, for $0 \le \alpha < \alpha_1$,

$$
\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha(-\tau\Psi_j^\lambda + \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(-\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) \\
& +\alpha\mu \sum_{N(P_g^0)} |-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \alpha\mu\tau \\
& -\alpha\mu \sum_{N(P_=^0)} \min(-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) \\
& +\alpha\mu \sum_{N(M^0)} |-\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]| + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

For any $\tau \in (t_l^2[j], t_{l+1}^2[j])$, $l = 0 : k_j^2$,

$$\begin{aligned} i \in N(T^0): \quad \text{neg}[-\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j]] &= \pi_i^l[j] \\ i \in N(P_g^0): \quad \text{sign}[-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \pi_i^l[j] \\ i \in N(P_=^0): \quad \text{neg}[-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \pi_i^l[j] \\ i \in N(M^0): \quad \text{sign}[-\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]] &= \pi_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w + \alpha d) = p_\mu(w) + \alpha(-\tau r_\lambda^l[j] + r_g^l[j]) + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{K}_l[j]$, then the coefficient of α is negative. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^2[j], t_{l+1}^2[j])$.

Therefore, if $\mathcal{K}_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_\lambda < 0 < \sigma_g$ where $|\sigma_\lambda| < |\sigma_g|$. However, if there exists some $l \in \{0 : k_j^2\}$ such that $\mathcal{K}_l^*[j] \neq \emptyset$, then for any $\tau \in \mathcal{K}_l^*$, a direction d satisfying $\mathcal{A}_B^T d = (-\tau e_j^\lambda + e_j^g)$ provides first order descent in p_μ from w. □

Definition 6.8 *For $j \in B(P_=^0)$:*

1. *Define breakpoints $\tau^3[j]$ as follows:*

$$i \in N(T^0): \quad \tau_i^3[j] = \begin{cases} \kappa_i^G[\lambda_j]/\kappa_i^G[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^3[j] = \begin{cases} \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \tau_i^3[j] = \begin{cases} \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^3[j] = \begin{cases} \kappa_i^c[\lambda_j]/\kappa_i^c[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise.} \end{cases}$$

2. *Let* k_j^3 *be the number of distinct breakpoints in the interval* $(0,1)$, *and let these distinct breakpoints, denoted* $t_l^3[j]$ *for* $l = 1, \cdots, k_j^3$, *be indexed to satisfy*

$$0 < t_1^3[j] < \cdots < t_{k_j^3}^3[j] < 1.$$

In addition, define the breakpoints $t_0^3[j] = 0$ *and* $t_{k_j^3+1}^3[j] = 1$.

3. *For each* $l = 0 : k_j^3$, *define* $\varphi^l[j]$ *as follows:*

$$i \in N(T^0): \quad \varpi_i^l[j] = \begin{cases} pos[\kappa_i^G[\lambda_j]] & \text{if } \tau_i^3[j] < t_l^3[j] \\ neg[\kappa_i^G[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \varpi_i^l[j] = \begin{cases} sign[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^3[j] < t_l^3[j] \\ -sign[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \varpi_i^l[j] = \begin{cases} pos[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^3[j] < t_l^3[j] \\ neg[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \varpi_i^l[j] = \begin{cases} sign[\kappa_i^c[\lambda_j]] & \text{if } \tau_i^3[j] < t_l^3[j] \\ -sign[\kappa_i^c[g_j]] & \text{otherwise.} \end{cases}$$

4. *For each* $l = 0 : k_j^3$, *define the quantities* $q_\lambda^l[j]$ *and* $q_g^l[j]$ *as follows:*

$$\begin{aligned} q_\lambda^l[j] &= \Psi_j^\lambda - \mu \sum_{N(T^0)} \varpi_i^l[j]\kappa_i^G[\lambda_j] - \mu \sum_{N(P_g^0)} \varpi_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_=^0)} \varpi_i^l[j]\kappa_i^g[\lambda_j] \\ &\quad -\mu \sum_{N(M^0)} \varpi_i^l[j]\kappa_i^c[\lambda_j] - \mu \\ q_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \varpi_i^l[j]\kappa_i^G[g_j] - \mu \sum_{N(P_g^0)} \varpi_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \varpi_i^l[j]\kappa_i^g[g_j] \\ &\quad -\mu \sum_{N(M^0)} \varpi_i^l[j]\kappa_i^c[g_j]. \end{aligned}$$

5. *For each $l = 0 : k_j^3$, define the intervals $\mathcal{N}_l[j]$ and $\mathcal{N}_l^*[j]$ as follows:*

$$\mathcal{N}_l[j] = \begin{cases} (-\infty, q_\lambda^l[j]/q_g^l[j]) & \text{if } q_g^l[j] > 0 \\ (q_\lambda^l[j]/q_g^l[j], \infty) & \text{if } q_g^l[j] < 0 \\ (-\infty, \infty) & \text{if } q_g^l[j] = 0 \text{ and } q_\lambda^l[j] > 0 \\ \emptyset & \text{otherwise,} \end{cases}$$

$$\mathcal{N}_l^*[j] = (t_l^3[j], t_{l+1}^3[j]) \cap \mathcal{N}_l[j].$$

Lemma 6.14 *If w is a degenerate, stationary point, then there exists a first order descent direction d for p_μ from w satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < 0 < \sigma_g$ and $|\sigma_\lambda| > |\sigma_g|$ if and only if there exists $l \in \{0 : k_j^3\}$ such that $\mathcal{N}_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $\mathcal{N}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_\lambda < 0 < \sigma_g$ and $|\sigma_\lambda| > |\sigma_g|$. Without loss of generality, assume that $\sigma_\lambda = -1$ and $\sigma_g = \tau$ for some $\tau \in (0, 1)$. Then, for $0 \leq \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) - \alpha(\Psi_j^\lambda - \tau \Psi_j^g) + \alpha\mu \sum_{N(T^0)} \max(\kappa_i^G[\lambda_j] - \tau\kappa_i^G[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]| + \alpha\mu \\ &\quad + \alpha\mu \sum_{N(P_=^0)} \max(\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \tau\kappa_i^c[g_j]| \\ &\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

For any $\tau \in (t_l^3[j], t_{l+1}^3[j])$, $l = 0 : k_j^3$,

$$\begin{aligned} i \in N(T^0): \quad \text{pos}[\kappa_i^G[\lambda_j] - \tau\kappa_i^G[g_j]] &= \varpi_i^l[j] \\ i \in N(P_g^0): \quad \text{sign}[\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]] &= \varpi_i^l[j] \\ i \in N(P_=^0): \quad \text{pos}[\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]] &= \varpi_i^l[j] \\ i \in N(M^0): \quad \text{sign}[\kappa_i^c[\lambda_j] - \tau\kappa_i^c[g_j]] &= \varpi_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w+\alpha d) \;=\; p_\mu(w) - \alpha(q^l_\lambda[j] - \tau q^l_g[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w,\alpha d).$$

If $\tau \in \mathcal{N}_l[j]$, then the coefficient of $(-\alpha)$ is positive. However, this value provides the first order rate of change in p_μ only for $\tau \in (t^3_l[j], t^3_{l+1}[j])$.

Therefore, if $\mathcal{N}^*_l = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_\lambda < 0 < \sigma_g$ and $|\sigma_\lambda| > |\sigma_g|$. However, if there exists some $l \in \{0 : k^3_j\}$ such that $\mathcal{N}^*_l[j] \neq \emptyset$, then for any $\tau \in \mathcal{N}^*_l$, a direction d satisfying $\mathcal{A}^T_B d = -(e^\lambda_j - \tau e^g_j)$ provides first order descent in p_μ from w. □

Definition 6.9 *For $j \in B(P^0_=)$:*

1. *Define breakpoints $\tau^4[j]$ as follows:*

$$i \in N(T^0): \quad \tau^4_i[j] \;=\; \begin{cases} -\;\kappa^G_i[g_j]/\kappa^G_i[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P^0_g): \quad \tau^4_i[j] \;=\; \begin{cases} -\;\kappa^g_i[g_j]/\kappa^g_i[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P^0_=): \quad \tau^4_i[j] \;=\; \begin{cases} -\;\kappa^g_i[g_j]/\kappa^g_i[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau^4_i[j] \;=\; \begin{cases} -\;\kappa^c_i[g_j]/\kappa^c_i[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise.} \end{cases}$$

2. *Let k^4_j be the number of distinct breakpoints in the interval $(0,1)$, and let these distinct breakpoints, denoted $t^4_l[j]$ for $l = 1, \cdots, k^4_j[j]$, be indexed to satisfy*

$$0 < t^4_1[j] < \cdots < t^4_{k^4_j}[j] < 1.$$

In addition, define the breakpoints $t^4_0[j] = 0$ and $t^4_{k^4_l+1}[j] = 1$.

3. *For each* $l = 0 : k_j^4$, *define* $\rho^l[j]$ *as follows:*

$$i \in N(T^0): \quad \rho_i^l[j] = \begin{cases} neg[\kappa_i^G[g_j]] & \text{if } \tau_i^4[j] < t_l^4[j] \\ neg[\kappa_i^G[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \rho_i^l[j] = \begin{cases} sign[\kappa_i^g[g_j]] & \text{if } \tau_i^4[j] < t_l^4[j] \\ sign[\kappa_i^g[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_{=}^0): \quad \rho_i^l[j] = \begin{cases} neg[\kappa_i^g[g_j]] & \text{if } \tau_i^4[j] < t_l^4[j] \\ neg[\kappa_i^g[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \rho_i^l[j] = \begin{cases} sign[\kappa_i^c[g_j]] & \text{if } \tau_i^4[j] < t_l^4[j] \\ sign[\kappa_i^c[\lambda_j]] & \text{otherwise.} \end{cases}$$

4. *For each* $l = 0 : k_j^4$, *define the quantities* $p_\lambda^l[j]$ *and* $p_g^l[j]$ *as follows:*

$$\begin{aligned} p_\lambda^l[j] &= \Psi_j^\lambda - \mu \sum_{N(T^0)} \rho_i^l[j]\kappa_i^G[\lambda_j] + \mu \sum_{N(P_g^0)} \rho_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_{=}^0)} \rho_i^l[j]\kappa_i^g[\lambda_j] \\ &\quad + \mu \sum_{N(M^0)} \rho_i^l[j]\kappa_i^c[\lambda_j] + \mu \\ p_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \rho_i^l[j]\kappa_i^G[g_j] + \mu \sum_{N(P_g^0)} \rho_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_{=}^0)} \rho_i^l[j]\kappa_i^g[g_j] \\ &\quad + \mu \sum_{N(M^0)} \rho_i^l[j]\kappa_i^c[g_j]. \end{aligned}$$

5. *For each* $l = 0 : k_j^4$, *define the intervals* $Q_l[j]$ *and* $Q_l^*[j]$ *as follows:*

$$\begin{aligned} Q_l[j] &= \begin{cases} (-\infty, -p_g^l[j]/p_\lambda^l[j]) & \text{if } p_\lambda^l[j] > 0 \\ (-p_g^l[j]/p_\lambda^l[j], \infty) & \text{if } p_\lambda^l[j] < 0 \\ (-\infty, \infty) & \text{if } p_\lambda^l[j] = 0 \text{ and } p_g^l[j] < 0 \\ \emptyset & \text{otherwise,} \end{cases} \\ Q_l^*[j] &= (t_l^4[j], t_{l+1}^4[j]) \cap Q_l[j]. \end{aligned}$$

Lemma 6.15 *If* w *is a degenerate, stationary point, then there exists a first order descent direction* d *for* p_μ *from* w *satisfying* $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ *for* $0 < \sigma_\lambda < \sigma_g$ *if and only if there exists* $l \in \{0 : k_j^4\}$ *such that* $Q_l^*[j] \neq \emptyset$.

Proof: Consider any direction d satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $0 < \sigma_\lambda < \sigma_g$. Without loss of generality, assume that $\sigma_g = 1$ and $\sigma_\lambda = \tau$ for some $\tau \in (0,1)$. Then, for $0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) + \alpha(\tau \Psi_j^\lambda + \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \alpha\mu\tau \\ &\quad - \alpha\mu \sum_{N(P_=^0)} \min(\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]| \\ &\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

For any $\tau \in (t_l^4[j], t_{l+1}^4[j])$, $l = 0 : k_j^4$,

$$\begin{aligned} i \in N(T^0): \quad \text{neg}[\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j]] &= \rho_i^l[j] \\ i \in N(P_g^0): \quad \text{sign}[\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \rho_i^l[j] \\ i \in N(P_=^0): \quad \text{neg}[\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \rho_i^l[j] \\ i \in N(M^0): \quad \text{sign}[\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]] &= \rho_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w + \alpha d) = p_\mu(w) + \alpha(\tau p_\lambda^l[j] + p_g^l[j]) + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{Q}_l[j]$, then the coefficient of α is negative. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^4[j], t_{l+1}^4[j])$.

Therefore, if $\mathcal{Q}_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $0 < \sigma_\lambda < \sigma_g$. However, if there exists some $l \in \{0 : k_j^4\}$ such that $\mathcal{Q}_l^*[j] \neq \emptyset$, then for any $\tau \in \mathcal{Q}_l^*$, a direction d satisfying $\mathcal{A}_B^T d = (\tau e_j^\lambda + e_j^g)$ provides first order descent in p_μ from w. □

Definition 6.10 *For $j \in B(P_=^0)$:*

1. Define breakpoints $\tau^5[j]$ as follows:

$$i \in N(T^0): \quad \tau_i^5[j] \;=\; \begin{cases} -\;\; \kappa_i^G[g_j]/\kappa_i^G[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^5[j] \;=\; \begin{cases} -\;\; \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \tau_i^5[j] \;=\; \begin{cases} -\;\; \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^5[j] \;=\; \begin{cases} -\;\; \kappa_i^c[g_j]/\kappa_i^c[\lambda_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise.} \end{cases}$$

2. *Let k_j^5 be the number of distinct breakpoints in the interval $(0,1)$, and let these distinct breakpoints, denoted t_l^5 be indexed to satisfy*

$$0 < t_1^5[j] < \cdots < t_{k_j^5}^5[j] < 1.$$

In addition, define the breakpoints $t_0^5[j] = 0$ and $t_{k_j^5+1}^5[j] = 1$.

3. *For each $l = 0 : k_j^5$, define $\varrho^l[j]$ as follows:*

$$i \in N(T^0): \quad \varrho_i^l[j] \;=\; \begin{cases} pos[\kappa_i^G[g_j]] & \textit{if } \tau_i^5[j] < t_l^5[j] \\ pos[\kappa_i^G[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \varrho_i^l[j] \;=\; \begin{cases} sign[\kappa_i^g[g_j]] & \textit{if } \tau_i^5[j] < t_l^5[j] \\ sign[\kappa_i^g[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \varrho_i^l[j] \;=\; \begin{cases} pos[\kappa_i^g[g_j]] & \textit{if } \tau_i^5[j] < t_l^5[j] \\ pos[\kappa_i^g[\lambda_j]] & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \varrho_i^l[j] \;=\; \begin{cases} sign[\kappa_i^c[g_j]] & \textit{if } \tau_i^5[j] < t_l^5[j] \\ sign[\kappa_i^c[\lambda_j]] & \textit{otherwise.} \end{cases}$$

4. *For each $l = 0 : k_j^5$, define the quantities $n_\lambda^l[j]$ and $n_g^l[j]$ as follows:*

$$n_\lambda^l[j] \;=\; \Psi_j^\lambda - \mu \sum_{N(T^0)} \varrho_i^l[j]\kappa_i^G[\lambda_j] - \mu \sum_{N(P_g^0)} \varrho_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_=^0)} \varrho_i^l[j]\kappa_i^g[\lambda_j]$$

$$
\begin{aligned}
& -\mu \sum_{N(M^0)} \varrho_i^l[j]\kappa_i^c[\lambda_j] \\
n_g^l[j] \;&=\; \Psi_j^g - \mu \sum_{N(T^0)} \varrho_i^l[j]\kappa_i^G[g_j] - \mu \sum_{N(P_g^0)} \varrho_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \varrho_i^l[j]\kappa_i^g[g_j] \\
& -\mu \sum_{N(M^0)} \varrho_i^l[j]\kappa_i^c[g_j] - \mu.
\end{aligned}
$$

5. *For each* $l = 0 : k_j^5$*, define the intervals* $\mathcal{R}_l[j]$ *and* $\mathcal{R}_l^*[j]$ *as follows:*

$$
\begin{aligned}
\mathcal{R}_l[j] \;&=\; \begin{cases} (-\infty, -n_g^l[j]/n_\lambda^l[j]) & \text{if } n_\lambda^l[j] < 0 \\ (-n_g^l[j]/n_\lambda^l[j], \infty) & \text{if } n_\lambda^l[j] > 0 \\ (-\infty, \infty) & \text{if } n_\lambda^l[j] = 0 \text{ and } n_g^l[j] > 0 \\ \emptyset & \text{otherwise,} \end{cases} \\
\mathcal{R}_l^*[j] \;&=\; (t_l^5[j], t_{l+1}^5[j]) \cap \mathcal{R}_l[j].
\end{aligned}
$$

Lemma 6.16 *If w is a degenerate, stationary point, then there exists a first order descent direction d for p_μ from w satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < \sigma_\lambda < 0$ if and only if there exists $l \in \{0 : k_j^5\}$ such that $\mathcal{R}_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < \sigma_\lambda < 0$. Without loss of generality, assume that $\sigma_g = -1$ and $\sigma_\lambda = -\tau$ for some $\tau \in (0,1)$. Then, for $0 \leq \alpha < \alpha_1$,

$$
\begin{aligned}
p_\mu(w + \alpha d) \;&=\; p_\mu(w) - \alpha(\tau\Psi_j^\lambda + \Psi_j^g) + \alpha\mu \sum_{N(T^0)} \max(\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) \\
& +\alpha\mu \sum_{N(P_g^0)} |\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \alpha\mu \\
& +\alpha\mu \sum_{N(P_=^0)} \max(\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]| \\
& +\frac{1}{2}\alpha^2\|d\|_2^2 \Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

For any $\tau \in (t_l^5[j], t_{l+1}^5[j])$, $l = 0 : k_j^5$,

$$\begin{aligned}
i \in N(T^0): \quad \text{pos}[\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j]] &= \varrho_i^l[j] \\
i \in N(P_g^0): \quad \text{sign}[\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \varrho_i^l[j] \\
i \in N(P_=^0): \quad \text{pos}[\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \varrho_i^l[j] \\
i \in N(M^0): \quad \text{sign}[\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]] &= \varrho_i^l[j],
\end{aligned}$$

and therefore

$$p_\mu(w + \alpha d) = p_\mu(w) - \alpha(\tau n_\lambda^l[j] + n_g^l[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{R}_l[j]$, then the coefficient of $(-\alpha)$ is positive. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^5[j], t_{l+1}^5[j])$.

Therefore, if $\mathcal{R}_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_\lambda < \sigma_g < 0$. However, if there exists some $l \in \{0 : k_j^5\}$ such that $\mathcal{R}_l^*[j] \neq \emptyset$, then for any $\tau \in \mathcal{R}_l^*$, a direction d satisfying $\mathcal{A}_B^T d = -(\tau e_j^\lambda + e_j^g)$ provides first order descent in p_μ from w. □

Definition 6.11 *For $j \in B(P_=^0)$:*

1. *Define breakpoints $\tau^6[j]$ as follows:*

$$i \in N(T^0): \quad \tau_i^6[j] = \begin{cases} \kappa_i^G[\lambda_j]/\kappa_i^G[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^6[j] = \begin{cases} \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \tau_i^6[j] = \begin{cases} \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^6[j] = \begin{cases} \kappa_i^c[\lambda_j]/\kappa_i^c[g_j] & \textit{if it lies in } (0,1) \\ 1 & \textit{otherwise.} \end{cases}$$

2. *Let* k_j^6 *be the number of distinct breakpoints in the interval* $(0, 1)$, *and let these distinct breakpoints, denoted* $t_l^6[j]$ *for* $l = 1, \cdots, k_j^6$, *be indexed to satisfy*

$$0 < t_2^1[j] < \cdots < t_{k_j^6}^6[j] < 1.$$

In addition, define the breakpoints $t_0^6[j] = 0$ *and* $t_{k_j^6+1}^6[j] = 1$.

3. *For each* $l = 0 : k_j^6$, *define* $\sigma^l[j]$ *as follows:*

$$i \in N(T^0): \quad \sigma_i^l[j] = \begin{cases} neg[\kappa_i^G[\lambda_j]] & \text{if } \tau_i^6[j] < t_l^6[j] \\ pos[\kappa_i^G[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \sigma_i^l[j] = \begin{cases} sign[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^6[j] < t_l^6[j] \\ -sign[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_{=}^0): \quad \sigma_i^l[j] = \begin{cases} neg[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^6[j] < t_l^6[j] \\ pos[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \sigma_i^l[j] = \begin{cases} sign[\kappa_i^{\underline{c}}[\lambda_j]] & \text{if } \tau_i^6[j] < t_l^6[j] \\ -sign[\kappa_i^{\underline{c}}[g_j]] & \text{otherwise.} \end{cases}$$

4. *For each* $l = 0 : k_j^6$, *define the quantities* $m_\lambda^l[j]$ *and* $m_g^l[j]$ *as follows:*

$$\begin{aligned} m_\lambda^l[j] &= \Psi_j^\lambda - \mu \sum_{N(T^0)} \sigma_i^l[j]\kappa_i^G[\lambda_j] + \mu \sum_{N(P_g^0)} \sigma_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_{=}^0)} \sigma_i^l[j]\kappa_i^g[\lambda_j] \\ &\quad + \mu \sum_{N(M^0)} \sigma_i^l[j]\kappa_i^{\underline{c}}[\lambda_j] \\ m_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \sigma_i^l[j]\kappa_i^G[g_j] + \mu \sum_{N(P_g^0)} \sigma_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_{=}^0)} \sigma_i^l[j]\kappa_i^g[g_j] \\ &\quad + \mu \sum_{N(M^0)} \sigma_i^l[j]\kappa_i^{\underline{c}}[g_j] - \mu. \end{aligned}$$

5. *For each $l = 0 : k_j^6$, define the intervals $S_l[j]$ and $S_l^*[j]$ as follows:*

$$S_l[j] = \begin{cases} (-\infty, m_\lambda^l[j]/m_g^l[j]) & \text{if } m_g^l[j] < 0 \\ (m_\lambda^l[j]/m_g^l[j], \infty) & \text{if } m_g^l[j] > 0 \\ (-\infty, \infty) & \text{if } m_g^l[j] = 0 \text{ and } m_\lambda^l[j] < 0 \\ \emptyset & \text{otherwise,} \end{cases}$$
$$S_l^*[j] = (t_l^6[j], t_{l+1}^6[j]) \cap S_l[j].$$

Lemma 6.17 *If w is a degenerate, stationary point, then there exists a first order descent direction d for p_μ from w satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < 0 < \sigma_\lambda$ and $|\sigma_g| < |\sigma_\lambda|$ if and only if there exists $l \in \{0 : k_j^6\}$ such that $S_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < 0 < \sigma_\lambda$ and $|\sigma_g| < |\sigma_\lambda|$. Without loss of generality, assume that $\sigma_\lambda = 1$ and $\sigma_g = -\tau$ for some $\tau \in (0, 1)$. Then, for $0 \leq \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) + \alpha(\Psi_j^\lambda - \tau\Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[\lambda_j] - \tau\kappa_i^G[g_j], 0) \\ &\quad + \alpha\mu \sum_{N(P_g^0)} |\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]| + \alpha\mu\tau \\ &\quad - \alpha\mu \sum_{N(P_=^0)} \min(\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \tau\kappa_i^c[g_j]| \\ &\quad + \frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

For any $\tau \in (t_l^6[j], t_{l+1}^6[j])$, $l = 0 : k_j^6$,

$$\begin{aligned} i \in N(T^0): \quad \operatorname{neg}[\kappa_i^G[\lambda_j] - \tau\kappa_i^G[g_j]] &= \sigma_i^l[j] \\ i \in N(P_g^0): \quad \operatorname{sign}[\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]] &= \sigma_i^l[j] \\ i \in N(P_=^0): \quad \operatorname{neg}[\kappa_i^g[\lambda_j] - \tau\kappa_i^g[g_j]] &= \sigma_i^l[j] \\ i \in N(M^0): \quad \operatorname{sign}[\kappa_i^c[\lambda_j] - \tau\kappa_i^c[g_j]] &= \sigma_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w+\alpha d) \;=\; p_\mu(w) + \alpha(m_\lambda^l[j] - \tau m_g^l[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w,\alpha d).$$

If $\tau \in S_l[j]$, then the coefficient of α is negative. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^6[j], t_{l+1}^6[j])$.

Therefore, if $S_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_g < 0 < \sigma_\lambda$ where $|\sigma_g| < |\sigma_\lambda|$. However, if there exists some $l \in \{0 : k_j^6\}$ such that $S_l^*[j] \neq \emptyset$, then for any $\tau \in S_l^*$, a direction d satisfying $\mathcal{A}_B^T d = (e_j^\lambda - \tau e_j^g)$ provides first order descent in p_μ from w. □

Definition 6.12 *For $j \in B(P_=^0)$:*

1. *Define breakpoints $\tau^7[j]$ as follows:*

$$i \in N(T^0): \quad \tau_i^7[j] \;=\; \begin{cases} \kappa_i^G[g_j]/\kappa_i^G[\lambda_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^7[j] \;=\; \begin{cases} \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \tau_i^7[j] \;=\; \begin{cases} \kappa_i^g[g_j]/\kappa_i^g[\lambda_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^7[j] \;=\; \begin{cases} \kappa_i^c[g_j]/\kappa_i^c[\lambda_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise.} \end{cases}$$

2. *Let k_j^7 be the number of distinct breakpoints in the interval $(0,1)$, and let these distinct breakpoints, denoted $t_l^7[j]$ for $l = 1, \cdots, k_j^7$, be indexed to satisfy*

$$0 < t_1^7[j] < \cdots < t_{k_j^7}^7[j] < 1.$$

In addition, define the breakpoints $t_0^7[j] = 0$ and $t_{k_j^7+1}^7[j] = 1$.

3. For each $l = 0 : k_j^7$, define $\varsigma^l[j]$ as follows:

$$i \in N(T^0): \quad \varsigma_i^l[j] = \begin{cases} pos[\kappa_i^G[g_j]] & \text{if } \tau_i^7[j] < t_l^7[j] \\ neg[\kappa_i^G[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \varsigma_i^l[j] = \begin{cases} sign[\kappa_i^g[g_j]] & \text{if } \tau_i^7[j] < t_l^7[j] \\ -sign[\kappa_i^g[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_=^0): \quad \varsigma_i^l[j] = \begin{cases} pos[\kappa_i^g[g_j]] & \text{if } \tau_i^7[j] < t_l^7[j] \\ neg[\kappa_i^g[\lambda_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \varsigma_i^l[j] = \begin{cases} sign[\kappa_i^c[g_j]] & \text{if } \tau_i^7[j] < t_l^7[j] \\ -sign[\kappa_i^c[\lambda_j]] & \text{otherwise.} \end{cases}$$

4. For each $l = 0 : k_j^7$, define the quantities $\ell_\lambda^l[j]$ and $\ell_g^l[j]$ as follows:

$$\begin{aligned} \ell_\lambda^l[j] &= \Psi_j^\lambda - \mu \sum_{N(T^0)} \varsigma_i^l[j]\kappa_i^G[\lambda_j] - \mu \sum_{N(P_g^0)} \varsigma_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_=^0)} \varsigma_i^l[j]\kappa_i^g[\lambda_j] \\ &\quad -\mu \sum_{N(M^0)} \varsigma_i^l[j]\kappa_i^c[\lambda_j] \\ \ell_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \varsigma_i^l[j]\kappa_i^G[g_j] - \mu \sum_{N(P_g^0)} \varsigma_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \varsigma_i^l[j]\kappa_i^g[g_j] \\ &\quad -\mu \sum_{N(M^0)} \varsigma_i^l[j]\kappa_i^c[g_j] - \mu. \end{aligned}$$

5. For each $l = 0 : k_j^7$, define the intervals $\mathcal{U}_l[j]$ and $\mathcal{U}_l^*[j]$ as follows:

$$\mathcal{U}_l[j] = \begin{cases} (-\infty, \ell_g^l[j]/\ell_\lambda^l[j]) & \text{if } \ell_\lambda^l[j] > 0 \\ (\ell_g^l[j]/\ell_\lambda^l[j], \infty) & \text{if } \ell_\lambda^l[j] < 0 \\ (-\infty, \infty) & \text{if } \ell_\lambda^l[j] = 0 \text{ and } \ell_g^l[j] > 0 \\ \emptyset & \text{otherwise,} \end{cases}$$

$$\mathcal{U}_l^*[j] = (t_l^7[j], t_{l+1}^7[j]) \cap \mathcal{U}_l[j].$$

Lemma 6.18 *If w is a degenerate, stationary point, then there exists a first order descent direction d for p_μ from w satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < 0 < \sigma_\lambda$ and $|\sigma_g| > |\sigma_\lambda|$ if and only if there exists $l \in \{0 : k_j^7\}$ such that $\mathcal{U}_l^*[j] \neq \emptyset$.*

Proof: Consider any direction d satisfying $\mathcal{A}_B^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $\sigma_g < 0 < \sigma_\lambda$ and $|\sigma_g| > |\sigma_\lambda|$. Without loss of generality, assume that $\sigma_g = -1$ and $\sigma_\lambda = \tau$ for some $\tau \in (0, 1)$. Then, for $0 \le \alpha < \alpha_1$,

$$\begin{aligned} p_\mu(w + \alpha d) &= p_\mu(w) - \alpha(-\tau\Psi_j^\lambda + \Psi_j^g) + \alpha\mu \sum_{N(T^0)} \max(-\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j], 0) \\ &\quad +\alpha\mu \sum_{N(P_g^0)} |-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]| + \alpha\mu \\ &\quad +\alpha\mu \sum_{N(P_=^0)} \max(-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] - \tau\kappa_i^c[g_j]| \\ &\quad +\frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d). \end{aligned}$$

For any $\tau \in (t_l^\tau[j], t_{l+1}^\tau[j])$, $l = 0 : k_j^\tau$,

$$\begin{aligned} i \in N(T^0): &\quad \text{pos}[-\tau\kappa_i^G[\lambda_j] + \kappa_i^G[g_j]] &= \varsigma_i^l[j] \\ i \in N(P_g^0): &\quad \text{sign}[-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \varsigma_i^l[j] \\ i \in N(P_=^0): &\quad \text{pos}[-\tau\kappa_i^g[\lambda_j] + \kappa_i^g[g_j]] &= \varsigma_i^l[j] \\ i \in N(M^0): &\quad \text{sign}[-\tau\kappa_i^c[\lambda_j] + \kappa_i^c[g_j]] &= \varsigma_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w + \alpha d) = p_\mu(w) - \alpha(-\tau\ell_\lambda^l[j] + \ell_g^l[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{U}_l[j]$, then the coefficient of $(-\alpha)$ is positive. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^\tau[j], t_{l+1}^\tau[j])$.

Therefore, if $\mathcal{U}_l^* = \emptyset$ for all l, then first order descent in p_μ is not possible for any $\sigma_g < 0 < \sigma_\lambda$ and $|\sigma_g| > |\sigma_\lambda|$. However, if there exists some $l \in \{0 : k_j^\tau\}$ such that $\mathcal{U}_l^*[j] \neq \emptyset$, then for any $\tau \in \mathcal{U}_l^*$, a direction d satisfying $\mathcal{A}_B^T d = -(-\tau e_j^\lambda + e_j^g)$ provides first order descent in p_μ from w. □

Definition 6.13 *For $j \in B(P_=^0)$:*

1. *Define breakpoints $\tau^8[j]$ as follows:*

$$i \in N(T^0): \quad \tau_i^8[j] = \begin{cases} -\ \kappa_i^G[\lambda_j]/\kappa_i^G[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \tau_i^8[j] = \begin{cases} -\ \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(P_{=}^0): \quad \tau_i^8[j] = \begin{cases} -\ \kappa_i^g[\lambda_j]/\kappa_i^g[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \tau_i^8[j] = \begin{cases} -\ \kappa_i^c[\lambda_j]/\kappa_i^c[g_j] & \text{if it lies in } (0,1) \\ 1 & \text{otherwise.} \end{cases}$$

2. *Let k_j^8 be the number of distinct breakpoints in the interval $(0,1)$, and let these distinct breakpoints, denoted $t_l^8[j]$ for $l = 1, \cdots, k_j^8$, be indexed to satisfy*

$$0 < t_1^8[j] < \cdots < t_{k_j^8}^8[j] < 1.$$

In addition, define the breakpoints $t_0^8[j] = 0$ and $t_{k_j^8+1}^8[j] = 1$.

3. *For each $l = 0 : k_j^8$, define $\chi^l[j]$ as follows:*

$$i \in N(T^0): \quad \chi_i^l[j] = \begin{cases} neg[\kappa_i^G[\lambda_j]] & \text{if } \tau_i^8[j] < t_l^8[j] \\ neg[\kappa_i^G[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_g^0): \quad \chi_i^l[j] = \begin{cases} sign[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^8[j] < t_l^8[j] \\ sign[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(P_{=}^0): \quad \chi_i^l[j] = \begin{cases} neg[\kappa_i^g[\lambda_j]] & \text{if } \tau_i^8[j] < t_l^8[j] \\ neg[\kappa_i^g[g_j]] & \text{otherwise} \end{cases}$$

$$i \in N(M^0): \quad \chi_i^l[j] = \begin{cases} sign[\kappa_i^c[\lambda_j]] & \text{if } \tau_i^8[j] < t_l^8[j] \\ sign[\kappa_i^c[g_j]] & \text{otherwise.} \end{cases}$$

4. *For each $l = 0 : k_j^8$, define the quantities $t_\lambda^l[j]$ and $t_g^l[j]$ as follows:*

$$t_\lambda^l[j] = \Psi_j^\lambda - \mu \sum_{N(T^0)} \chi_i^l[j]\kappa_i^G[\lambda_j] + \mu \sum_{N(P_g^0)} \chi_i^l[j]\kappa_i^g[\lambda_j] - \mu \sum_{N(P_{=}^0)} \chi_i^l[j]\kappa_i^g[\lambda_j]$$

$$
\begin{aligned}
&+\mu \sum_{N(M^0)} \chi_i^l[j]\kappa_i^c[\lambda_j] \\
t_g^l[j] &= \Psi_j^g - \mu \sum_{N(T^0)} \chi_i^l[j]\kappa_i^G[g_j] + \mu \sum_{N(P_s^0)} \chi_i^l[j]\kappa_i^g[g_j] - \mu \sum_{N(P_=^0)} \chi_i^l[j]\kappa_i^g[g_j] \\
&+\mu \sum_{N(M^0)} \chi_i^l[j]\kappa_i^c[g_j] + \mu.
\end{aligned}
$$

5. *For each* $l = 0 : k_j^8$, *define the intervals* $\mathcal{V}_l[j]$ *and* $\mathcal{V}_l^*[j]$ *as follows:*

$$
\begin{aligned}
\mathcal{V}_l[j] &= \begin{cases} (-\infty, -t_\lambda^l[j]/t_g^l[j]) & \text{if } t_g^l[j] > 0 \\ (-t_\lambda^l[j]/t_g^l[j], \infty) & \text{if } t_g^l[j] < 0 \\ (-\infty, \infty) & \text{if } t_g^l[j] = 0 \text{ and } t_\lambda^l[j] < 0 \\ \emptyset & \text{otherwise,} \end{cases} \\
\mathcal{U}_l^*[j] &= (t_l^8[j], t_{l+1}^8[j]) \cap \mathcal{V}_l[j].
\end{aligned}
$$

Lemma 6.19 *If* w *is a degenerate, stationary point, then there exists a first order descent direction* d *for* p_μ *from* w *satisfying* $\mathcal{A}_\mathcal{B}^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ *for* $0 < \sigma_g < \sigma_\lambda$ *if and only if there exists* $l \in \{0 : k_j^8\}$ *such that* $\mathcal{V}_l^*[j] \neq \emptyset$.

Proof: Consider any direction d satisfying $\mathcal{A}_\mathcal{B}^T d = \sigma_\lambda e_j^\lambda + \sigma_g e_j^g$ for $0 < \sigma_g < \sigma_\lambda$. Without loss of generality, assume that $\sigma_\lambda = 1$ and $\sigma_g = \tau$ for some $\tau \in (0, 1)$. Then, for $0 \leq \alpha < \alpha_1$,

$$
\begin{aligned}
p_\mu(w + \alpha d) &= p_\mu(w) + \alpha(\Psi_j^\lambda + \tau \Psi_j^g) - \alpha\mu \sum_{N(T^0)} \min(\kappa_i^G[\lambda_j] + \tau\kappa_i^G[g_j], 0) \\
&+\alpha\mu \sum_{N(P_s^0)} |\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]| + \alpha\mu\tau \\
&-\alpha\mu \sum_{N(P_=^0)} \min(\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j], 0) + \alpha\mu \sum_{N(M^0)} |\kappa_i^c[\lambda_j] + \tau\kappa_i^c[g_j]| \\
&+\frac{1}{2}\alpha^2 \|d\|_2^2 \Omega_{p_\mu}(w, \alpha d).
\end{aligned}
$$

For any $\tau \in (t_l^{\rm s}[j], t_{l+1}^{\rm s}[j])$, $l = 0 : k_j^{\rm s}$,

$$\begin{aligned} i \in N(T^0): \quad \text{neg}[\kappa_i^G[\lambda_j] + \tau\kappa_i^G[g_j]] &= \chi_i^l[j] \\ i \in N(P_g^0): \quad \text{sign}[\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]] &= \chi_i^l[j] \\ i \in N(P_=^0): \quad \text{neg}[\kappa_i^g[\lambda_j] + \tau\kappa_i^g[g_j]] &= \chi_i^l[j] \\ i \in N(M^0): \quad \text{sign}[\kappa_i^c[\lambda_j] + \tau\kappa_i^c[g_j]] &= \chi_i^l[j], \end{aligned}$$

and therefore

$$p_\mu(w + \alpha d) = p_\mu(w) + \alpha(t_\lambda^l[j] + \tau t_g^l[j]) + \frac{1}{2}\alpha^2\|d\|_2^2\Omega_{p_\mu}(w, \alpha d).$$

If $\tau \in \mathcal{V}_l[j]$, then the coefficient of α is negative. However, this value provides the first order rate of change in p_μ only for $\tau \in (t_l^{\rm s}[j], t_{l+1}^{\rm s}[j])$.

Therefore, if $\mathcal{V}_l^- = \emptyset$ for all l, then first order descent in p_μ is not possible for any $0 < \sigma_g < \sigma_\lambda$. However, if there exists some $l \in \{0 : k_j^{\rm s}\}$ such that $\mathcal{V}_l^-[j] \neq \emptyset$, then for any $\tau \in \mathcal{V}_l^-$, a direction d satisfying $\mathcal{A}_B^T d = (e_j^\lambda + \tau e_j^g)$ provides first order descent in p_μ from w. □

6.7 First Order Optimality Conditions

Corollary 6.20 *If w is a degenerate, first order point of the penalty function p_μ, then the following conditions must all be satisfied.*

$$\begin{array}{llll} j \in B(T^0): & \Psi_j^G - \mu u_j^G \le \mu & \textit{and} & \Psi_j^G + \mu v_j^G \ge 0, \\ j \in P_\lambda^0: & \Psi_j^\lambda - \mu u_j^G \le \mu & \textit{and} & \Psi_j^\lambda + \mu v_j^\lambda \ge -\mu, \\ j \in B(P_g^0): & \Psi_j^g - \mu u_j^g \le \mu & \textit{and} & \Psi_j^g + \mu v_j^g \ge -\mu, \\ j \in P_=^-: & \Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} \le \mu & \textit{and} & \Psi_j^{g-\lambda} + \mu v_j^{g-\lambda} \ge 0, \\ j \in P_=^+: & \Psi_j^{g-\lambda} - \mu u_j^{g-\lambda} \le -\mu & \textit{and} & \Psi_j^{g-\lambda} + \mu v_j^{g-\lambda} \ge 0, \end{array}$$

$$
\begin{array}{llll}
j \in B(P^0_=): & \Psi^\lambda_j - \mu u^\lambda_j \leq \mu & \textit{and} & \Psi^\lambda_j + \mu v^\lambda_j \geq 0, \\
\textit{and} & \Psi^g_j - \mu u^g_j \leq \mu & \textit{and} & \Psi^g_j + \mu v^g_j \geq 0, \\
\textit{and} & \Psi^\lambda_j + \Psi^g_j - \mu u^{\lambda+g}_j \leq \mu & \textit{and} & \Psi^\lambda_j + \Psi^g_j + \mu v^{\lambda+g}_j \geq -\mu, \\
\textit{and} & \Psi^\lambda_j - \Psi^g_j - \mu u^{\lambda-g}_j \leq \mu & \textit{and} & \Psi^\lambda_j - \Psi^g_j + \mu v^{\lambda-g}_j \geq -\mu, \\
j \in N(P^0_=): & \Psi^\lambda_j - \mu u^\lambda_j - \mu \max(1, \kappa^g_j[\lambda_j]) \leq 0 & \textit{and} & \\
 & \Psi^\lambda_j + \mu v^\lambda_j + \mu |\min(1, \kappa^g_j[\lambda_j])| \geq 0, & & \\
j \in B(M^0): & \Psi^c_j - \mu u^c_j \leq \mu & \textit{and} & \Psi^c_j + \mu v^c_j \geq -\mu,
\end{array}
$$

and for all $j \in B(P^0_=)$,

$$
\begin{array}{rcll}
\mathcal{J}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^1_j, \\
\mathcal{K}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^2_j, \\
\mathcal{N}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^3_j, \\
\mathcal{Q}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^4_j, \\
\mathcal{R}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^5_j, \\
\mathcal{S}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^6_j, \\
\mathcal{U}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^7_j, \\
\mathcal{V}^*_l[j] & = & \emptyset & \textit{for} \quad l = 0 : k^8_j.
\end{array}
$$

Proof: Follows immediately from the results presented in the previous section. □

Note that, in the absence of degeneracy, the matrix κ is empty and the conditions listed above reduce to the first order necessary conditions previously defined in Corollary 3.10 and Definition 3.3.4 for a nondegenerate minimum point of the penalty function. Also, if a set of multipliers Ψ satisfying the nondegenerate necessary optimality conditions are defined at a degenerate point of p_μ, then the above conditions are automatically satisfied for some choice of $\mathcal{A}_B$. Therefore, the κ multipliers need only be calculated when the multipliers Ψ do not satisfy the nondegenerate necessary conditions and do not define a descent direction for the penalty function.

6.8 Implemented Version

Our degeneracy resolving technique has been described for exact activities. With appropriate modifications to the activity sets, this technique is implemented for ϵ-activities exactly as it is described here. It has been verified that the technique will either find a descent direction or verify that the degenerate point satisfies the necessary optimality conditions. With the introduction of ϵ-activities, an additional possibility arises. As described in Section 4.5, it is now possible that the current point has been misclassified as close to a stationary point of the penalty function. As a result, the dropping direction returned by the degeneracy resolver may not be a descent direction for the model function. In this case, the algorithm will reduce the values of ϵ and Λ, and continue from the reclassified point.

6.9 Conclusion

In this chapter, the problems inherent in applying traditional techniques to resolving degeneracy in the penalty function p_μ have been explained. In addition, we have derived a set of first order necessary conditions for a degenerate solution of p_μ. These conditions apply regardless of the actual choice of the activity basis matrix $\mathcal{A}_B$ as long as the rules described in this chapter for constructing $\mathcal{A}_B$ are followed.

The calculation of the vector Ψ and matrix κ for any specific choice of $\mathcal{A}_B$ and $\mathcal{A}_N$ and the checking of all the conditions listed above are finite processes. In any particular instance of degeneracy in the penalty function, the method of perturbation or the technique of Ryan and Osborne may resolve the problem more quickly than the multiplier method, or may fail completely. Unlike these techniques, the multiplier method will always resolve degeneracy in p_μ in a finite number of steps.

Chapter 7

Testing results

7.1 Introduction

The penalty function algorithm described in Chapters 3 and 4 for finding a solution of the one level problem BP_C was implemented in Matlab. The degeneracy resolving technique described in Chapter 6 was also implemented. The code was tested on a set of bilevel programming problems found in the literature, as well as several original nonlinear problems. In this section, I will discuss the test problems and describe how the testing was performed. The remainder of the chapter is concerned with the presentation and analysis of the results.

7.1.1 Test Problems

The test problems are listed in Appendix A along with their sources and known solutions. The majority of the problems were found by examining the bilevel programming literature. These problems have generally been used to illustrate proposed algorithms for bilevel problems, and therefore, most are quite small, with only a few upper and lower level variables and constraints. Also, since most of the literature to date is concerned with linear bilevel problems, the majority of the problems presented in the literature were linear problems. While several quadratic problems were found, only three nonquadratic nonlinear problems were located. Therefore, several new nonlinear problems, some with unknown global solutions, were developed.

In [28], Calamai and Vicente described a method for generating random linear and quadratic bilevel problems with known global solutions. The problems are separable, but with a simple matrix transformation, nonseparable quadratic bilevel problems can be generated. Seven untransformed and five transformed problems, of varying sizes, are used in the testing process.

For all but one of the test problems from the literature, at least one global solution is known. In addition, because most of the problems are small, it was usually possible to analyze them to determine other local and global solutions. The goal of our research was to develop an algorithm which could be used to find solutions of bilevel programming problems. Therefore, the implemented code was tested on problems with known solutions so that its performance could be evaluated with regard to this criteria. In addition, it was necessary to introduce new nonlinear problems so that more results could be obtained.

7.1.2 Code and Algorithm Parameters

The algorithms were implemented and tested using Matlab version 4.2c. Throughout the testing process, emphasis was placed on ensuring that the code was performing correctly rather than on improving the speed of convergence. Therefore, little attempt was made to find the "best" initial values of the algorithm variables or parameters. The values used for the starting point w^0 and the initial penalty parameter μ^0 are considered in the next section. The remaining algorithm variables and parameters used the same initial values for all test problems, as indicated below. These values should perform reasonably well for well-scaled problems.

The following initial values are used throughout the testing process.

- ϵ^0, the initial activity tolerance, has value 0.1.
- Λ^0, the initial closeness tolerance, has value 0.2.
- Δ^0, the initial trust region radius, has value $\min(\Delta^{max}, 0.1 \times \|\gamma^0\|_2)$, where γ^0 is the gradient of the differentiable part of the penalty function at w^0. However, if $\|\gamma^0\|_2 < 1$, then $\Delta^0 = 1$.

The algorithm parameters, as specified in the statement of Algorithm 4.1, are assigned the following values. Some of the values are based on the numerical experience of other researchers, and the remaining values (for example, b_{max}, Δ_{max}, and it_{max}) seem reasonable for the problems being solved.

- b_1, the tolerance for a successful trust region direction, has value 10^{-4}.
- b_2, the tolerance for a very successful trust region direction, has value 0.75.
- r_1, the sufficient decrease tolerance, has value 10^{-4}.
- b_{max}, the iteration count corresponding to unboundedness, has value 4.
- Δ_{max}, the maximum allowable trust region radius, has value 20.
- μ_{max}, the maximum allowable value of the penalty parameter, has value 10^6.
- it_{max}, the maximum allowable number of iterations before unsuccessful termination of the algorithm, has value 500 for nonlinear problems, and value 100 for the other problems. The larger value was used for the nonlinear problems because they are generally more difficult to solve.

A point is accepted as a second order point of the penalty function if the following conditions are satisfied at the current point.

- $\|Z^T\gamma\|_2 \leq 5 \times 10^{-5}$,
- all activities have values α satisfying $|\alpha| \leq 5 \times 10^{-5}$,
- the multipliers lie very close to the optimal ranges for second order points stated in Corollary 6.20. For example, if the optimal range is $[\rho_1, \rho_2]$, where $\rho_1 \neq 0$, then the actual multipliers must lie in the range $[(1-\xi)\rho_1, (1+\xi)\rho_2]$ where $\xi = 0.01$. If $\rho_1 = 0$, then the actual multiplier must lie in the range $[-\xi, (1+\xi)\rho_2]$, and
- the reduced Hessian $Z^T HZ$ is positive semidefinite.

A point is accepted as a second order point of the one level form of the bilevel problem if it satisfies the above conditions for a second order point of p_μ and it is essentially feasible for BP_C, i.e. if

$$|p_\mu(w) - F(w)| \leq 5 \times 10^{-5}\mu.$$

7.1.3 Testing Process

The test problems listed in Appendix A were each run for ten different combinations of starting point w^0 and initial penalty parameter μ^0. These values were chosen for all the test problems, and no attempt was made to find the best starting point for individual problems. The tested values, corresponding to the results presented later in the chapter, are listed in Table 7.1, where $q = n + m + p$ is the number of variables in the penalty function. Recall that n is the number of upper level variables, m is the number of lower level variables, and p is the number of lower level constraints.

#	w^0	μ^0
1	$zeros(q)$	1
2	$ones(q)$	1
3	$-ones(q)$	1
4	$5 \times ones(q)$	1
5	$-10 \times ones(q)$	1
6	$randn(q)$	10
7	$5 \times randn(q)$	100
8	$-5 \times randn(q)$	1000
9	$10 \times randn(q)$	100
10	$-10 \times randn(q)$	10

Table 7.1: Starting Values

Note that

- $zeros(q)$ is the zero vector in $\mathbb{R}^q$,
- $ones(q)$ is the vector in $\mathbb{R}^q$ consisting of all ones, and
- $randn(q)$ is a vector in $\mathbb{R}^q$ consisting of q normally distributed random values.

The term "problem instance" will be used to refer to a test problem in combination with one of the above (w^0, μ^0) combinations.

7.2 Presentation of the Results

The test problems listed in Appendix A are divided into four groups: linear, quadratic, generated, and more general nonlinear bilevel problems. Accordingly, the results are presented in four separate tables. In each of the result tables, the following information is presented.

- The name of the problem, as listed in Appendix A, generally corresponds to the initials of the authors of the source paper. It is stated in the column titled "Prob".
- The number of variables in the one level form of the bilevel problem is indicated in the column titled "q".
- The average number of iterations until termination of the implemented algorithm is indicated in the column titled "#". This is the average over all ten instances, regardless of whether the algorithm actually converges or not.
- The average number of times that the penalty parameter was increased for each of the ten problem instances is indicated in the column titled "μ^+". Recall that the parameter μ is increased if the algorithm indicates that p_μ is becoming unbounded, or if the algorithm converges to a point satisfying the necessary conditions for a second order point of p_μ which is infeasible for the one level problem. Each time μ is increased, it is multiplied by the factor 10.
- The average number of times that the degeneracy resolving routine was invoked for each of the ten problem instances is indicated in the column titled "δ". The routine, as described in Chapter 6, is invoked if the algorithm encounters a degenerate point at which some of the multipliers Ψ are out of kilter but the calculated dropping direction is not a descent direction.
- The algorithm can terminate in several different ways.
  - The final point satisfies the necessary conditions for a second order point of the one level form of the bilevel problem and the objective function value at

the point matches the objective function value of the known global solution of the bilevel problem. The number of times during the ten test trials that the algorithm terminates at such a point, classified as a global solution, is indicated under the outcome column titled "G".

- The final point satisfies the necessary conditions for a second order point of the one level form of the bilevel problem, but the objective function value at the point exceeds the objective function value of the known global solution of the bilevel problem. The number of times during the ten test trials that the algorithm terminates at such a point, classified as a local solution, is indicated under the outcome column titled "L".
- The final point satisfies the necessary conditions for a second order point of p_μ for $\mu = \mu_{\max}$, but is not feasible for the one level form of the bilevel problem. The number of times during the ten test trials that the algorithm terminates at such a point, classified as a truly infeasible solution, is indicated under the outcome column titled "I".
- The algorithm fails to converge within $it_{\max}$ iterations. The number of times that this outcome was observed is indicated in the column titled "M".
- The algorithm terminates because p_μ appeared to be unbounded for $\mu = \mu_{\max}$. The number of times that this outcome was observed is indicated in the column titled "U".

7.3 Results and Comments

Throughout the analysis of the results, the performance of our algorithm will not be compared to existing bilevel problem algorithms. There are several reasons for this decision. Our algorithm has been designed to solve nonlinear bilevel problems, and therefore, is not expected to perform as well for special forms of the problem as algorithms designed for those forms. For example, linear bilevel algorithms exploit the special properties of linear bilevel problems and include actions to find global, rather than local, solutions. Our algorithm does not include such special steps. In addition, at the time of testing, there are no other algorithms in the literature for which extensive test results have been presented for nonlinear bilevel problems. Finally, the performance of the algorithm is likely far from optimal because no analysis has been performed to determine the best starting values of the algorithm variables and parameters, as noted on page 231.

7.3.1 Linear Problems

The testing process for the linear bilevel test problems, as summarized in Table 7.2, illustrated the following points.

- The algorithm identified a global solution of the bilevel problem for 18 of the 23 problems with known global solutions. More specifically, it found the global solution in 105 or 46% of the 230 associated problem instances. For the ten problem instances for BCC5, for which the global solution is not known, the algorithm terminated half of the time at the best local solution provided by the authors.
- The algorithm identified a local or global solution of the bilevel problem for all 24 problems, or 153 (64%) of the 240 problem instances.
- A truly infeasible point of the penalty function was identified in 21 of the 24 problems or 107 (36%) of the problem instances.
- The degeneracy resolving routine was invoked for 7 of the problems, or 15 (6%) of the 240 problems instances. In several cases, the routine was invoked multiple times for a single problem instance.

- A solution of the penalty function was successfully identified for all problem instances.
- The algorithm converged within 44 iterations for all problem instances.

The algorithm performed extremely well in identifying local or global solutions of the linear bilevel test problems, as there were no problems for which a solution was not identified. Also, the performance of the algorithm in identifying global solutions was quite good. The use of multiple starting points was a helpful tool in identifying global solutions for a large majority of the problems. Even without special action within the algorithm for specific starting points, global solutions were identified for one-third of all problem instances. This behavior is very encouraging. Of course, in general, such solutions are probably not globally convergent.

Aside from their use in finding global solutions, multiple starting points must be used within the bilevel algorithm due to the possibility of identifying truly infeasible solutions. The number of problem instances which terminated at a truly infeasible point justifies their use.

Although only a small percentage of problem instances encountered a point for which the degeneracy resolving routine was invoked, one-quarter of the problems did require its use. This illustrates the importance of the degeneracy resolving technique developed specifically for the bilevel penalty function.

Although all the problems in this group have a small number of variables, the three highest average number of iterations to convergence correspond to the three largest problems, as indicated by the value of q. The problems with more than 10 variables are the only problems which, on average, required more than 20 iterations. This suggests that the number of iterations may increase with the problem size.

Prob	q	#	μ^+	δ	G	L	I
AW1	8	13.3	0.5	0	1	3	6
B1	8	12.0	0.4	0	5	0	5
B2	7	10.1	0.5	0.5	6	1	3
B3	7	12.3	0.5	0	7	0	3
BAB1	6	8.8	0.2	0.2	5	0	5
BAB2	7	13.1	0.5	0	4	0	6
BCC1	8	9.8	0.2	0	6	0	4
BCC2	7	9.2	0.3	0	0	5	5
BCC3	5	7.9	0	0	10	0	0
BCC4	8	12.4	0.4	0.1	8	0	2
BCC5	16	30.0	1.4	0		7	3
BF1	8	12.4	1.2	0.3	6	0	4
BF2	7	10.4	1.0	0.2	2	2	6
BK1	8	10.2	0	0.1	0	7	3
BK2	10	13.9	0.6	0.4	9	0	1
CF1	5	11.0	0.5	0	9	0	1
CT1	11	26.4	1.2	0	0	10	0
D1	8	12.2	0	0	10	0	0
D2	6	7.9	0	0	4	0	6
F1	6	7.8	0.4	0	2	5	3
HJS1	11	24.7	1.2	0	0	6	4
HJS2	7	11.5	0.5	0	0	1	9
HSW1	6	11.2	0.5	0	3	1	6
TMV1	7	14.6	0.8	0	8	0	2

Table 7.2: Results for Linear Problems

7.3.2 Quadratic Problems

The test results summarized in Table 7.3 illustrate the following points.

- The algorithm identified a global solution for eleven of the twelve quadratic bilevel test problems, or 63 (53%) of the 120 problem instances.
- A local or global solution is identified for all problems, or 97 (81%) of the 120 problem instances.
- A truly infeasible solution was located for 8 of the test problems, or 23 (19%) of the problem instances.
- The degeneracy resolving routine was invoked for seven of the test problems, or 37 (31%) of the problems instances.
- A solution of the penalty function was successfully identified for all problem instances.
- The algorithm converged within 26 iterations for all problem instances.

The algorithm performed exceptionally well in identifying the global and local solutions of this set of quadratic bilevel problems. While this may, in part, be due to the simplicity of the test problems, it is still very encouraging to see for our general nonlinear bilevel problem algorithm.

The number of truly infeasible points encountered by the algorithm even in these cases again justifies the use of multiple problem instances in the testing process. Similarly, the high number of degenerate points encountered show the necessity of the degeneracy resolving routine. The overall results also indicate the effectiveness of the routine.

Prob	q	#	μ^+	δ	G	L	I
AS1	5	7.8	0	0.4	3	0	7
AS2	10	12.2	0.5	0	4	6	0
B4	6	10.0	0.6	0	4	3	3
BF3	8	14.7	0.9	0.5	9	1	0
BF4	8	15.7	1.2	2.7	9	0	1
GS1	5	7.8	0	0	9	0	1
IA1	10	13.2	0.5	0	4	6	0
SG1	10	15.4	0.5	0	5	4	1
VSJ1	5	7.1	0.4	0.4	5	2	3
VSJ2	5	7.3	0.7	0.4	4	2	4
YZ1	4	4.5	0	0.6	7	0	3
YZZ2	4	7.7	0.5	1.2	0	10	0

Table 7.3: Results for Quadratic Problems

Once again, the largest problems, as indicated by the value of q, generally require the highest average number of iterations to convergence. The five largest problems (with $q \geq 8$) require more than 12 iterations on average, while none of the smaller problems require more than 10 iterations. However, since the problems are so small, this difference may not truly be significant.

7.3.3 Generated Problems

The testing results for the set of generated problems are summarized in Table 7.4, and illustrate the following points.

- A global solution of the bilevel problem is correctly identified for 10 of the 13 test problems, or 30 (23%) of the 130 problem instances.
- A local solution of the bilevel problem is correctly identified for all 13 test problems, or 88 (68%) of the problem instances.
- A truly infeasible solution of the penalty function was identified for 12 test problems, or 42 (32%) of the problem instances.
- Seven of the problems, or 11 (8%) of the problem instances, required the use of the degeneracy resolving routine.
- A solution of the penalty function was successfully identified for all problem instances.
- The algorithm converged within 63 iterations for all problem instances, and in less than 50 iterations for all but two.

Again, the algorithm performed quite well in identifying global and local solutions of these test problems. These problems, by their nature, generally have more local solutions than the previous cases. So, while the high number of local solutions encountered was not surprising, the strong performance of the algorithm in identifying global solutions was an encouraging result, and was supported by the performance of the algorithm on the previous set of quadratic problems.

The number of iterations required for convergence is generally higher for this set of test problems than for the previous two sets of problems, reflecting the larger size of the problems and their increased complexity. Note that the larger untransformed and transformed problems do not always require a higher average number of iterations for convergence. While the largest problems seem to generally require more iterations for convergence, this is not necessarily true for specific problems.

Prob	q	#	μ^+	δ	G	L	I
U1	25	14.5	1.4	0.6	2	5	3
U2	30	14.0	0.8	0	1	6	3
U3	50	18.0	1.4	0.1	7	3	0
U4	55	19.9	1.0	0	5	3	2
U5	80	24.8	1.3	0.5	0	6	4
U6	125	25.6	1.1	0.4	2	6	2
U7	90	20.1	1.2	0	1	6	3
U8	100	15.8	2.2	0	1	3	6
T1	12	11.5	1.3	0	1	4	5
T2	25	21.4	3.1	0.3	0	4	6
T3	30	14.4	1.0	0	0	8	2
T4	30	13.7	0.6	0.1	9	0	1
T5	55	32.0	2.2	0.3	1	4	5

Table 7.4: Results for Generated Problems

7.3.4 Nonlinear Problems

Only three nonlinear bilevel test problems were found in the literature. As this number was insufficient for our testing purposes, several new problems were designed. The global solutions of some of these problems, as indicated in Appendix A, are not known. Note that the problems tested here are small, in terms of the number of variables in the penalty function. The time and space limitations imposed by Matlab and, in particular, the current implementation of the algorithm make solving much larger problems impractical.

The results for the problems from the literature (problems B5, EB1 and YZZ1), along with the new problems, are summarized in Table 7.5. The performance of our algorithm in solving these nonlinear bilevel problems cannot be judged relative to any other solution techniques. A literature search found no comparable results for other methods.

- For the thirteen problems with known global solutions, the algorithm identified a global solution in 27 (20%) of the 130 problem instances. Note that the algorithm identified a global solution in a similar percentage of cases for the generated problems, which are the most complicated of the other problems tested here.

- A global or local solution was identified for all the problems, and in 95 (63%) of the 150 problem instances. Again, this compares very well with the observed results for both the linear and generated problems.

- A truly infeasible solution was identified for 12 of the problems, and in 24 (16%) of the problem instances.

- The degeneracy resolving technique was invoked for five of the test problems, or 19 (13%) of the problem instances.

- For 26 (17%) of the problem instances (corresponding to 12 of the test problems), the algorithm failed to converge to a solution of the penalty function within $it_{\max}$ iterations. For the nonlinear problems, the maximum iteration count allowed was $it_{\max} = 500$.

- The problem instance was found to be unbounded in 11 (7%) of the problem instances, or for five of the problems.

- Of the 124 problem instances which converged to a local or global solution, a truly infeasible point, or were judged to be unbounded, 20 (16%) required more that 100 iterations.

  - One problem instance (B5) converged after 498 iterations.

– Six problem instances (Cc1, Cc3, two of Ce1, two of Ce3) required between 300 and 399 iterations.

– Six problem instances (two of Cb3, one of Cc1, Cc3, Ce1 and Cg1) required between 200 and 299 iterations.

– Seven problem instances (two of Cb3, one of Cc1, CC3, two of Ce3, and one of Cg3) required between 100 and 199 iterations.

In the nonlinear results table, the two numbers $i_1(i_2)$ under the column titled "$\#(\#_c)$" indicate i_1, the average number of iterations to termination over all ten problem instances, and i_2, the average number of iterations to termination when the it_{max} instances are omitted.

Two types of results were observed for the nonlinear problems which were not observed for any of the previous test problems: unboundedness (column "U") and failure to converge within it_{max} iterations (column "M"). In the original discussion of the penalty function technique in Chapter 3, unboundedness was discussed as a possible outcome of the penalty function technique, even if the problem being solved was not unbounded. The technique used to detect unboundedness is described in Section 3.8. It is likely that a more detailed check for unboundedness could be developed which would eliminate some of the unboundedness outcomes. However, this issue will not be investigated further here.

As noted above, there were 26 instances in which the algorithm failed to converge or to reach an unboundedness decision within $it_{max} = 500$ iterations. The behavior of the algorithm for these instances is discussed below. Two different patterns of behavior accounted for most of these outcomes. Note that these patterns of behavior were generally established within the first 100 to 200 iterations of the algorithm.

1. The algorithm made steady, but very small progress towards a stationary point of the penalty function. Over these iterations, the iterates were all type one points,

Prob	q	$\#(\#_c)$	μ^+	δ	G	L	I	U	M
B5	8	120.1 (77.9)	0.6	0	7	2	0	0	1
EB1	8	11.8	1.2	0.4	6	0	4	0	0
YZZ1	4	110.5 (13.1)	0.1	0	3	2	3	0	2
Ca1	3	8.9	0.5	0	3	7	0	0	0
Ca3	9	13.0	0.5	0	2	8	0	0	0
Cb1	4	135.3 (44.1)	2.3	0		4	1	3	2
Cb3	12	235.0 (121.4)	3.0	0		2	1	4	3
Cc1	5	135.3 (94.8)	1.4	4.6	2	5	1	1	1
Cc3	15	238.8 (116.9)	1.9	9.4	0	5	1	1	3
Cd1	3	109.3 (11.6)	0.2	0	1	6	1	0	2
Cd3	9	211.7 (19.5)	0.3	0	0	5	1	0	4
Ce1	4	212.5 (140.7)	0.6	0.5	1	5	2	0	2
Ce3	12	197.9 (97.4)	2.1	16.2	2	3	3	0	2
Cg1	4	139.0 (48.8)	0.8	0	0	6	2	0	2
Cg3	12	149.7 (62.1)	2.4	0	0	2	4	2	2

Table 7.5: Results for Nonlinear Problems

and the generalized Cauchy direction provided acceptable (though usually relatively poor) trust region descent. In some cases, the direction provided very good descent. Generally, however, the trust region radius Δ was relatively small when the pattern was established, and was not increased or decreased very much over the subsequent iterations.

This was the pattern of behavior observed for the it_{max} outcomes for problems YZZ1 (two instances), Cb1 (two instances), Cb3 (three instances), Cc3 (one instance), Cd1 (2 instances), Cd3 (one instance), Ce1 (two instances), Cg1 (one instance) and Cg3 (one instance).

2. After the trust region radius was reduced to a relatively small value, the iterate was classified as close to a stationary, non-first order point. A dropping direction was calculated, and it provided a small, but acceptable descent in the penalty function. Because the step taken, while a reasonable trust region direction, was small, the dropped ϵ-activities remained active. The next iteration again tried to drop the same set of activities, and the process was repeated.

   This pattern of behavior was observed for Cc1 (one instance), Cc3 (one instance), Cd3 (two instances), Ce3 (two instances) and Cg1 (one instance).

The remaining four cases fail for different, but related reasons. For all cases, after the preliminary stages of the algorithm, the iterates were close to a stationary point of the penalty function. However, the algorithm did not always recognize this property, as described below.

- In problem B5, the algorithm actually approached a second order point of the penalty function relatively quickly. The iterate was correctly classified as a type four point, and the full Newton step was attempted. However, the step failed. Note that the step would have been accepted if the current value of the penalty parameter μ had been significantly smaller. The algorithm tolerances ϵ and Λ were reduced, and the iterate was then misclassified as far from stationarity. All subsequent descent directions were generalized Cauchy steps from type one points. Acceptable decrease was observed, but the decrease was actually quite small.

- For the remaining instance of problem Cc3, the iterates were actually close to stationarity, but because of previous decreases of ϵ and Λ, some were classified as type one points, and some were classified as type two points. In the former situations, the generalized Cauchy step provided good, but not very good, descent. In the latter situations, the dropping step was successful, but the dropped activities remained ϵ-active. In essence, both patterns of behavior described above were observed here.

- For the remaining case of Cd3, the iterates were close to a first, non-second order point, and were correctly classified as type three points. The calculated directions of negative curvature provided very small, but still acceptable descent in the penalty function. The trust region radius was near its minimum value. It is possible that a direction of negative curvature like those discussed in Section 3.5.3 would have been beneficial in this case.

- For the remaining case, one instance of Cg3, an iterate was correctly classified as close to a second order point. A full Newton step was attempted and accepted. However, as a result of the step, one of the activities was no longer considered active at the next iterate. This new point was incorrectly classified as far from a stationary point, and very small generalized Cauchy directions provided acceptable descent in the penalty function, as discussed above.

The nonconvergent problem instances illustrate the need for multiple starting values when using our algorithm to solve bilevel problems.

The number of iterations required for convergence to a solution or to detect unboundedness is greater for these nonlinear problems than it is for the linear, quadratic and generated problems. This is to be expected due to the difficulty inherent in the nonlinear problem when compared to the simpler forms of the problems. In some cases, the increase is quite large.

Recall that the value of it_{max} was increased for the nonlinear problems to 500 from 100. This action was beneficial since the algorithm converged for more problem instances than it would have with the smaller value of the algorithm parameter. However, at the same time, it results in the average number of iterations required until termination of the algorithm to be skewed upwards by the instances which require more iterations. Table 7.6 presents summarized information in the spirit of Table 7.5, and corresponds to the performance of the algorithm when the value of it_{max} is reduced to 200. The number of instances which converged to a local or global solution are grouped together in the table under the column "L". Note that the average number of iterations presented is calculated by ignoring the it_{max} outcomes.

Prob	$\#_c$	L	I	U	M
B5	25.4	8	0	0	2
EB1	11.8	6	4	0	0
YZZ1	13.1	5	3	0	2
Ca1	8.9	10	0	0	0
Ca3	13.0	10	0	0	0
Cb1	44.1	4	1	3	2
Cb3	66.6	2	0	3	5
Cc1	45.6	6	1	0	3
Cc3	42.8	4	1	0	5
Cd1	11.6	7	1	0	2
Cd3	19.5	5	1	0	4
Ce1	12.6	5	0	0	5
Ce3	82.7	4	2	0	4
Cg1	22.4	5	2	0	3
Cg3	62.1	2	4	2	2

Table 7.6: Results for Nonlinear Problems when $it_{max} = 200$

The new table illustrates the following points.

- The algorithm locates a local or global solution for all the problems, and for 83 (55%) of the problem instances. This is compared to 63% for the larger value of it_{max}.
- The algorithm converges to a truly infeasible solution of the penalty function in 20 (13 %) of the instances, compared to 16% previously.
- The algorithm concludes that 8 (5%) of the problem instances are unbounded, down slightly from 7% previously.
- The algorithm terminates unsuccessfully after $it_{max} = 200$ iterations for 39 (26%) of the problem instances, compared to 17% after $it_{max} = 500$ iterations.

In addition to these points, note that the average number of iterations is, with a few exceptions, well within the range to be expected, based on the results presented for the linear, quadratic, and generated test problems, combined with the increased difficulty of solving the nonlinear problems. Of course, this came at a cost of fewer convergent problem instances. However, in terms of the amount of computing work in the extra 300 iterations, it may actually be preferable to use the smaller value of it_{max}. An alternate approach to consider is to define another problem instance whenever the current instance fails to converge within the smaller value of it_{max}.

7.4 Overall Comments

The algorithm performs quite well overall on the four sets of test problems. While the results of the nonlinear test problems may seem somewhat disappointing on first glance due to the number of nonconvergent instances, the algorithm actually performed very well in achieving the stated goal of locating local and global solutions of both the nonlinear bilevel problems and the "simpler" forms of the bilevel problem.

Consider the following points resulting from the analysis of the test results.

- The use of multiple problem instances in solving a specific problem is justified by the results, due to the abundance of truly infeasible solutions and of local solutions.
- Analyzing the individual problem to choose a starting point and initial penalty parameter may improve the performance of the algorithm in locating solutions, and may reduce the number of problem instances to consider.
- The implemented version of the degeneracy resolving technique is very useful and necessary.
- As expected, it appears that larger problems require more iterations before convergence, and each iteration generally requires more work than for smaller problems.
- The results presented for the linear, quadratic, and generated problems would not have been significantly different if the value of $it_{\max}$ had been reduced to 50 from its tested value of 100, as just two instances would have failed to converge. The results presented in Table 7.6 illustrate that, while the convergence results for the nonlinear problems are improved if $it_{\max}$ takes value 500 rather than 200, the improvement may not be significant enough to the user to justify the extra computational work involved.

  A "best" value for $it_{\max}$, as with most other algorithm parameters, depends on the individual bilevel problem and the user's objectives. This indicates that further study of these parameters may result in a better overall performance of the algorithm.

Chapter 8

Conclusion

The thesis concludes with a list of contributions of the research along with an indication of possible further work in the field.

8.1 Contributions

We believe this work provides several significant contributions to the field of bilevel programming in the areas of algorithm development, algorithm convergence, and testing.

- Algorithm Development.

  An algorithm has been designed for the nonlinear bilevel programming problem. As explained in Chapter 2, most algorithms in the literature have been described for simpler forms of the bilevel problem. Our algorithm, which implements an exact ℓ_1 penalty function technique within a trust region framework, places few restrictions on the problem form.

  The combination of the penalty function and the trust region techniques is particularly appropriate to the bilevel problem. The penalty function technique is designed to handle the nondifferentiabilities of the compact form of the related problem, and assists in attaining both feasibility and optimality. The multiple starting points associated with the penalty function technique facilitates the discovery of both global and local solutions. At the same time, the trust region technique enables the algorithm to concentrate on a localized, simplified version of the penalty function. This approach is beneficial in handling the nonconvexities of the bilevel problem, as well as assisting overall convergence.

In addition, the algorithm includes a new technique, proven to work both in theory and in practice, for resolving degeneracy in the penalty function. This is particularly significant because traditional degeneracy resolving techniques were inappropriate for this problem.

- Algorithm Convergence.

  Under a set of assumptions standard to convergence analysis and a few assumptions specific to the bilevel problem, the proposed algorithm is proven to converge to a strict second order point of the penalty function for the compact form of the bilevel problem. These assumptions are stronger than would be applied to the problem in practice, but it is significant to note the strong theoretical nature underlying our technique. It provides a basis for its effectiveness for the more general problem.

  Note that, under an appropriate constraint qualification, as stated in Chapter 2, if the bilevel problem is convex, it is equivalent to its related compact form. In this case, the penalty function technique solves the bilevel problem directly. However, the convexity assumption on the bilevel problem is stronger than generally desired. For the more general case, the solution of the penalty function may still be a solution of the related compact problem, and hence of the bilevel problem itself.

- Testing.

  The numerical results presented here are the most extensive found to date for the nonlinear problem. They verify the effectiveness of the algorithm in identifying local and global solutions of the bilevel problem, both in the simpler and more general forms. The collection of test problems can be used for comparison purposes for newer algorithms.

8.2 Further Work

For each of the areas of contributions to the study of bilevel programming, there are some issues of research which remain open.

- Algorithm Development.

  There are two issues currently unresolved in the development of the algorithm.

  - It is possible, as described in Section 3.5.3, that the current iterate does not satisfy the necessary second order conditions for a solution because the reduced Hessian is indefinite. However, a direction of negative curvature may fail to provide descent in the model or penalty functions. As explained in the text, there are directions of descent at the point, but a practical technique for identifying such directions is required. While this situation was not encountered much during the testing process and was therefore not investigated further, it would be useful to resolve this issue.
  - As described in Section 7.3.4, a more detailed technique for identifying when the penalty function is becoming unbounded within the trust region framework would likely improve the observed performance of the algorithm.

- Algorithm Convergence.

  As described above, a set of assumptions, some stronger than others, are imposed on the problem to prove the theoretical convergence of the algorithm. It is of particular interest to determine if any of the assumptions specific to the bilevel problem (stated in Assumptions 5.8) can be removed and convergence still proven. This would match the theoretical results more closely with the practical convergence of the algorithm.

- Testing.

  The testing results provided for this algorithm are more complete than for other techniques. However, it would be useful to have more results for larger nonlinear bilevel problems. It would also be interesting to solve some practical applications whose solutions are not known *a priori*. Due to time and space limitations, this would require a new implementation of the algorithm outside the Matlab environment.

  In addition, a thorough analysis of the algorithm tolerances and parameters, both for all problems in general and for individual problems, would likely result in improved numerical performance.

Appendix A

Test Problems

The following problems have been found in the literature, or are originated here. The name attached to the problem signifies the original source. I have included any known solutions with the problem statement. In a few cases, the solutions given by the authors did not match those found by the algorithm. In case of conflict, the solution that could be verified (both by using the implemented algorithm and by analyzing the bilevel problem) is the one listed. If the authors gave a different solution which could not be verified, then it is not listed below. Included with the solution is F^*, the value of the upper level objective function at the solution.

We recognize that, with the exception of the problems generated by the technique of Calamai-Vicente as described below, the problems presented here are very small. This reflects the complexity of these classes of problems.

A.1 Linear Problems

1. Problem: AW1 from [6].

$$\begin{array}{lllll} \min_{x,y} & -(x+3y) \\ \text{s.t.} & x \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \min_{y} & 3y \\ & \quad \text{s.t.} & 10 \leq x+2y \leq 38 \\ & & -18 \leq x-2y \leq 6 \\ & & 2x-y \leq 21 \\ & & y \geq 0 \end{array}$$

Solutions:

(a) Global: $x^* = 16$, $y^* = 11$, $F^* = -49$.

(b) Local: $x^* = 0$, $y^* = 5$, $F^* = -15$.

2. Problem: B1 from [8].

$$\begin{array}{llll} \min_{x,y} & -2x_1 + x_2 + \frac{1}{2}y_1 \\ \text{s.t.} & 2x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \\ \text{and} & y \text{ solves} \\ & \text{LLP}(x): \min_{y} & -(4y_1 - y_2) \\ & \quad \text{s.t.} & 2x_1 - y_1 + y_2 \geq 2\frac{1}{2} \\ & & x_1 - 3x_2 + y_2 \leq 2 \\ & & y_1, y_2 \geq 0 \end{array}$$

Solutions:

(a) Global: $x^* = (1,0)$, $y^* = (\frac{1}{2}, 1)$, $F^* = -1\frac{3}{4}$.

3. **Problem: B2 from [10].**

$$\begin{array}{lll} \min_{x,y} & x+y & \\ \text{s.t.} & x \geq 0 \text{ and } y \text{ solves} & \\ & \text{LLP}(x) \quad \min_{y} & -y \\ & \qquad\qquad \text{s.t.} & x+\frac{1}{2}y \geq 2 \\ & & \frac{1}{4}x - y \geq -2 \\ & & x+\frac{1}{2}y \leq 8 \\ & & x-2y \leq 4 \\ & & y \geq 0 \end{array}$$

Solutions:

(a) Global: $x^* = \frac{8}{9}$, $y^* = 2\frac{2}{9}$, , $F^* = 3\frac{1}{9}$.

(b) Local: $x^* = 7\frac{1}{5}$, $y^* = 1\frac{3}{5}$, $F^* = 8\frac{4}{5}$.

4. **Problem: B3 from [10].**

$$\begin{array}{lll} \min_{x,y} & -(5x+y) & \\ \text{s.t.} & x \geq 0 \text{ and } y \text{ solves} & \\ & \text{LLP}(x): \quad \min_{y} & y \\ & \qquad\qquad \text{s.t.} & x+\frac{1}{2}y \geq 2 \\ & & \frac{1}{4}x - y \geq -2 \\ & & x+\frac{1}{2}y \leq 8 \\ & & x-2y \leq 4 \\ & & y \geq 0 \end{array}$$

Solutions:

(a) Global: $x^* = 7\frac{1}{5}$, $y^* = 1\frac{3}{5}$, $F^* = -37\frac{3}{5}$.

5. **Problem: BAB1 from [15].**

$$\begin{aligned}\min_{x,y} \quad & x+y \\ \text{s.t.} \quad & x \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x) \quad \min_y \quad y \\ & \qquad \text{s.t.} \quad \begin{aligned} 4x+3y &\geq 19 \\ x+2y &\leq 11 \\ 3x+y &\leq 13 \\ y &\geq 0 \end{aligned}\end{aligned}$$

Solutions:

(a) Global: $x^* = 1$, $y^* = 5$, $F^* = -6$.

6. **Problem: BAB2 from [15].**

$$\begin{aligned}\min_{x,y} \quad & -(1\frac{1}{2}x + 6y_1 + y_2) \\ \text{s.t.} \quad & 0 \leq x \leq 1 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \quad \min_y \quad -(y_1 + 5y_2) \\ & \qquad \text{s.t.} \quad \begin{aligned} x+3y_1+y_2 &\leq 5 \\ 2x+y_1+3y_2 &\leq 5 \\ y_1, y_2 &\geq 0 \end{aligned}\end{aligned}$$

Solutions:

(a) Global: $x^* = 1$, $y^* = (0,1)$, $F^* = -2\frac{1}{2}$.

7. **Problem: BCC1 from [18].**

$$\min_{x,y} \quad -y \quad \text{s.t. } y \text{ solves}$$

$$\begin{aligned} \text{LLP}(x): \min_{y} \quad & y \\ \text{s.t.} \quad & 2x + y \geq 6 \\ & x - 2y \geq -7 \\ & x + y \leq 9 \\ & 2x - y \leq 9 \\ & x + 3y \geq 8 \\ & y \leq 5 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = 1$, $y^* = 4$, $F^* = -4$.

(b) **Local:** $x^* = 6$, $y^* = 3$, $F^* = -3$.

8. **Problem:** BCC2 from [18].

$$\min_{x,y} \quad -y \quad \text{s.t. } y \text{ solves}$$

$$\begin{aligned} \text{LLP}(x): \min_{y} \quad & y \\ \text{s.t.} \quad & 10 \leq x + 2y \leq 38 \\ & -18 \leq x - 2y \leq 6 \\ & 2x - y \leq 21 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = 16$, $y^* = 11$, $F^* = -11$.

(b) **Local:** $x^* = -4$, $y^* = 7$, $F^* = -7$.

9. **Problem:** BCC3 from [18].

$$\min_{x,y} \quad x + y \quad \text{s.t. } y \text{ solves}$$

$$\begin{aligned} \text{LLP}(x): \quad \min_{y} \quad & -y \\ \text{s.t.} \quad 4x + 3y &\geq 19 \\ x + 2y &\leq 11 \\ 3x + y &\leq 13 \end{aligned}$$

Solutions:

(a) Global: $x^* = 4$, $y^* = 1$, $F^* = 5$.

(b) Local: $x^* = 1$, $y^* = 5$, $F^* = 6$.

10. Problem: BCC4 from [18].

$$\begin{aligned} \min_{x,y} \quad & x + y_1 \\ \text{s.t.} \quad & x \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \quad \min_{y} \quad -(y_1 - y_2) \\ & \qquad \text{s.t.} \quad x - \tfrac{1}{2}y_1 - \tfrac{1}{4}y_2 \geq 0 \\ & \qquad\qquad x + 12y_1 + \tfrac{1}{4}y_2 \leq 1 \\ & \qquad\qquad y_1, y_2 \geq 0 \\ & \qquad\qquad y_2 \leq 1 \end{aligned}$$

Solutions:

(a) Global: $x^* = 0$, $y^* = (0,0)$, $F^* = 0$.

11. Problem: BCC5 from [18].

$$\begin{aligned} \min_{x,y} \quad & -9x_1 - 3x_2 + 5x_3 - 2x_4 - 7y_1 - 30y_2 - 13y_3 \\ \text{s.t.} \quad & x_1, x_2, x_3, x_4 \geq 0 \\ \text{and} \quad & y \text{ solves} \end{aligned}$$

$$
\begin{aligned}
\text{LLP}(x):\ \min_{y}\quad & -(3y_1 - y_2 - 5y_3) \\
\text{s.t.}\quad & x_1 + 4x_2 + 4x_3 - x_4 - y_1 + y_2 + y_3 \le 1 \\
& x_1 - x_3 - 10x_4 + 2y_1 - 4y_2 + y_3 \ge -2 \\
& 3x_1 - 5x_2 + 4y_1 - 2y_2 - y_3 \le 2 \\
& x_1 + y_1 \ge -1 \\
& x_2 + y_2 \ge -1 \\
& x_3 + y_3 \ge -4 \\
& y_1, y_2, y_3 \ge 0
\end{aligned}
$$

Solutions:

(a) **Global:** Unknown.

(b) **Local:** $x^* = (0, \frac{6}{19}, 0, 0)$, $y^* = (\frac{29}{19}, \frac{24}{19}, 0)$, $F^* = -49.5263$.

12. **Problem:** BF1 from [12].

$$
\begin{aligned}
\min_{x \ge 0, y}\quad & -(2x_1 - x_2 - \frac{1}{2}y_1) \\
\text{s.t.}\quad & x_1 + x_2 \le 2 \\
& x_1, x_2 \ge 0 \text{ and } y \text{ solves} \\
& \text{LLP}(x):\ \min_{y}\quad -(4y_1 - y_2) \\
& \qquad \text{s.t.}\quad 2x_1 - y_1 + y_2 \ge 2\tfrac{1}{2} \\
& \qquad\qquad x_1 - 3x_2 + y_2 \le 2 \\
& \qquad\qquad y_1, y_2 \ge 0
\end{aligned}
$$

Solutions:

(a) **Global:** $x^* = (2, 0)$, $y^* = (3/2, 0)$, $F^* = -3\frac{1}{4}$.

13. **Problem:** BF2 from [12].

$$
\min_{x,y}\quad -(2x_1 - x_2 - 8y)
$$

$$
\begin{aligned}
\text{s.t.} \quad & x_1 + x_2 \leq 2 \\
& x_1, x_2 \geq 0 \text{ and } y \text{ solves} \\
& \text{LLP}(x): \min_y \quad -8y \\
& \qquad \text{s.t.} \quad \begin{aligned} x_1 + x_2 + y &\leq 3 \\ x_1 - y &\leq 0 \\ x_1 + x_2 - y &\geq -1 \\ y &\geq 0 \end{aligned}
\end{aligned}
$$

Solutions:

(a) Global: $x^* = (1,1)$, $y^* = 1$, $F^* = 7$.

(b) Local: $x^* = (0,0)$, $y^* = 1$, $F^* = 8$.

14. **Problem: BK1 [22].**

$$
\begin{aligned}
\min_{x,y} \quad & -y \\
\text{s.t.} \quad & x \geq 0 \text{ and } y \text{ solves} \\
& \text{LLP}(x): \min_y \quad y \\
& \qquad \text{s.t.} \quad \begin{aligned} 10 \leq x + 2y &\leq 38 \\ -18 \leq x - 2y &\leq 6 \\ 2x - y &\leq 21 \\ y &\geq 0 \end{aligned}
\end{aligned}
$$

Solutions:

(a) Global: $x^* = 16$, $y^* = 11$, $F^* = -11$.

(b) Local: $x^* = 0$, $y^* = 5$, $F^* = -5$.

15. **Problem: BK2 [22].**

$$
\min_{x,y} \quad -(x + y_2)
$$

$$\begin{aligned} \text{s.t.} \quad & x \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \min_{y \geq 0} \; -y_2 \\ & \qquad \text{s.t.} \quad \begin{array}{rcr} x + y_1 + y_2 & \leq & 3 \\ x + y_1 - y_2 & \geq & 1 \\ x - y_1 - y_2 & \geq & -1 \\ x - y_1 + y_2 & \leq & 1 \\ y_1, y_2 & \geq & 0 \\ y_2 & \leq & \frac{1}{2} \end{array} \end{aligned}$$

Solutions:

(a) Global: $x^* = \frac{3}{2}$, $y^* = (1, \frac{1}{2})$, $F^* = -2$.

(b) Global: $x^* = \alpha$, $y^* = (1, 2-\alpha)$ for any $\alpha \in (\frac{3}{2}, 2)$, $F^* = -2$.

(c) Global: $x^* = 2$, $y^* = (1, 0)$, $F^* = -2$.

16. Problem: CF1 from [32].

$$\begin{aligned} \min_{x,y} \quad & x - 4y \quad \text{s.t. } y \text{ solves} \\ & \text{LLP}(x): \min_{y} \; y \\ & \qquad \text{s.t.} \quad \begin{array}{rcr} 2x + 5y & \leq & 108 \\ 2x - 3y & \leq & -4 \\ x - y & \geq & 0 \end{array} \end{aligned}$$

Solutions:

(a) Global: $x^* = 19$, $y^* = 14$, $F^* = -37$.

17. Problem: CT1 from [31].

$$\min_{x,y} \quad -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$$

$$\begin{aligned} \text{s.t.} \quad & x_1, x_2 \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \min_{y \geq 0} \; y_1 + y_2 + 2y_3 \\ & \quad \text{s.t.} \quad \begin{aligned} 4x_1 - 2y_1 + 4y_2 - y_3 &\leq 2 \\ 4x_2 + 4y_1 - 2y_2 - y_3 &\leq 2 \\ y_1 - y_2 - y_3 &\geq -1 \\ y_1, y_2, y_3 &\geq 0 \end{aligned} \end{aligned}$$

Solutions:

(a) Global: $x^* = (0, \frac{9}{10})$, $y^* = (0, \frac{3}{5}, \frac{2}{5})$, $F^* = -29\frac{1}{5}$.

(b) Local: $x^* = (\frac{3}{2}, 0)$, $y^* = (1, 0, 2)$, , $F^* = -16$.

18. Problem: D1 [39].

$$\begin{aligned} \min_{x,y} \quad & -x \quad \text{s.t. } y \text{ solves} \\ & \text{LLP}(x): \min_{y} \; -y_2 \\ & \quad \text{s.t.} \quad \begin{aligned} 3x + 2y_1 - y_2 &\leq 5 \\ 3x + y_1 - 2y_2 &\geq -2 \\ 2y_1 + 3y_2 &\leq 18 \\ y_1, y_2 &\geq 0 \end{aligned} \end{aligned}$$

Solutions:

(a) Global: $x^* = 3\frac{2}{3}$, $y^* = (0, 6)$, $F^* = -3\frac{2}{3}$.

19. Problem: D2 from [39].

$$\min_{x,y} \quad x + y \quad \text{s.t. } y \text{ solves}$$

$$\begin{aligned} \text{LLP}(x): \min_{y} \quad & -y \\ \text{s.t.} \quad & 7x + y \geq 13 \\ & x + 3y \geq 9 \\ & 5x + 3y \leq 33 \\ & 5x + 7y \leq 47 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = 6$, $y^* = 1$, $F^* = 7$.

(b) **Global:** $x^* = 1$, $y^* = 6$, $F^* = 7$.

20. **Problem:** F1 from [42].

$$\begin{aligned} \min_{x,y} \quad & -y \\ \text{s.t.} \quad & x \geq 0 \text{ and } y \text{ solves} \\ & \quad \text{LLP}(x): \min_{y} \quad y \\ & \quad \text{s.t.} \quad x + y \geq 1 \\ & \quad \phantom{\text{s.t.} \quad} x - y \leq 1 \\ & \quad \phantom{\text{s.t.} \quad} 0 \leq y \leq 2 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = 3$, $y^* = 2$, $F^* = -2$.

(b) **Local:** $x^* = 0$, $y^* = 1$, $F^* = -1$.

21. **Problem:** HJS1 from [48].

$$\begin{aligned} \min_{x,y} \quad & -(8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3) \\ \text{s.t.} \quad & x_1 + 2x_2 - y_3 \leq 1.3 \\ \text{and} \quad & x_1, x_2 \geq 0 \text{ and } y \text{ solves} \end{aligned}$$

$$
\begin{aligned}
\text{LLP}(x): \quad \min_{y} \quad & 2y_1 + y_2 + 2y_3 \\
\text{s.t.} \quad & 4x_1 - 2y_1 + 4y_2 - y_3 \le 2 \\
& 4x_2 + 4y_1 - 2y_2 - y_3 \le 2 \\
& y_1 - y_2 - y_3 \ge -1 \\
& y_1, y_2, y_3 \ge 0
\end{aligned}
$$

Solutions:

(a) Global: $x^* = (\frac{1}{2}, \frac{4}{5})$, $y^* = (0, \frac{1}{5}, \frac{4}{5})$, $F^* = -18\frac{2}{5}$.

(b) Local: $x^* = (\frac{1}{2}, \frac{2}{5})$, $y^* = (0, 0, 0)$, $F^* = -5\frac{3}{5}$.

22. Problem: HJS2 from [47].

$$
\begin{aligned}
\min_{x,y} \quad & x + 5y \\
\text{s.t.} \quad & x \ge 0 \text{ and } y \text{ solves} \\
& \text{LLP}(x): \quad \min_{y} \quad -y \\
& \qquad \text{s.t.} \quad 3x - 2y \ge -6 \\
& \qquad\qquad x + 4y \le 48 \\
& \qquad\qquad x - 5y \le 9 \\
& \qquad\qquad x + y \ge 8 \\
& \qquad\qquad y \ge 0
\end{aligned}
$$

Solutions:

(a) Global: $x^* = 2$, $y^* = 6$, $F^* = 32$.

(b) Local: $x^* = 30\frac{2}{3}$, $y^* = 4\frac{1}{3}$, $F^* = 52\frac{1}{3}$.

23. Problem: HSW1 from [49].

$$
\min_{x,y} \quad x + 5y \quad \text{s.t. } y \text{ solves}
$$

$$\begin{aligned} \text{LLP}(x): \quad \min_{y} \quad & -y \\ \text{s.t.} \quad 3x - 2y &\geq -6 \\ 3x + 4y &\leq 48 \\ 2x - 5y &\leq 9 \\ x + y &\geq 8 \\ y &\geq 0 \end{aligned}$$

Solutions:

(a) Global: $x^* = 12$, $y^* = 3$, $F^* = 27$.

(b) Local: $x^* = 2$, $y^* = 6$, $F^* = 32$.

24. Problem: TMV1 from [65].

$$\begin{aligned} \min_{x \geq 0, y} \quad & 3x_1 + 2x_2 + y_1 + y_2 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 \leq 4 \\ & x_1, x_2, x_3, x_4 \geq 0 \\ \text{and} \quad & y \text{ solves} \\ & \text{LLP}(x): \quad \min_{y} \quad 4y_1 + y_2 \\ & \qquad \text{s.t.} \quad 3x_1 + 5x_2 + 6y_1 + 2y_2 \geq 15 \\ & \qquad\qquad y_1, y_2 \geq 0 \end{aligned}$$

Solutions:

(a) Global: $x^* = (0, 3)$, $y^* = (0, 0)$, $F^* = 6$.

A.2 Quadratic Problems

1. Problem: AS1 from [2].

$$\min_{x,y} \quad x^2 + (y - 10)^2$$

$$
\begin{aligned}
\text{s.t.} \quad & x \geq y \\
& 0 \leq x \leq 15 \\
\text{and} \quad & y \text{ solves} \\
& \quad \text{LLP}(x): \min_{y} \quad (x + 2y - 30)^2 \\
& \qquad\qquad\quad \text{s.t.} \quad x + y \leq 20 \\
& \qquad\qquad\qquad\quad\; 0 \leq y \leq 20
\end{aligned}
$$

Solutions:

(a) Global: $x^* = 10$, $y^* = 10$, $F^* = 100$.

2. Problem: AS2 from [2].

$$
\begin{aligned}
\min_{x,y} \quad & 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 \\
\text{s.t.} \quad & x_1 + x_2 + y_1 - 2y_2 \leq 40 \\
& 0 \leq x_1, x_2 \leq 50 \\
\text{and} \quad & y \text{ solves} \\
& \quad \text{LLP}(x): \min_{y} \quad (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2 \\
& \qquad\qquad\quad \text{s.t.} \quad -10 \leq y_1, y_2 \leq 20 \\
& \qquad\qquad\qquad\quad\; x_1 - 2y_1 \geq 10 \\
& \qquad\qquad\qquad\quad\; x_2 - 2y_2 \geq 10
\end{aligned}
$$

Solutions:

(a) Global: $x^* = (0, 0)$, $y^* = (-10, -10)$, $F^* = 0$.

(b) Local: $x^* = (25, 30)$, $y^* = (5, 10)$, $F^* = 5$.

3. Problem: B4 from [11].

$$
\min_{x,y} \quad (x - 5)^2 + (2y + 1)^2
$$

$$\text{s.t.} \quad x \geq 0 \text{ and } y \text{ solves}$$

$$\begin{aligned} \text{LLP}(x): \min_{y} \quad & (y-1)^2 - \tfrac{3}{2}xy \\ \text{s.t} \quad 3x - y &\geq 3 \\ x - \tfrac{3}{2}y &\leq 4 \\ x + y &\leq 7 \\ y &\geq 0 \end{aligned}$$

Solutions:

(a) Global: $x^* = 1$, $y^* = 0$, $F^* = 17$.

(b) Local: $x^* = 5$, $y^* = 2$, $F^* = 25$.

4. Problem: BF3 from [12].

$$\begin{aligned} \min_{x,y} \quad & x_1(2y_1 + 3y_2) + x_2(4y_1 + y_2) \\ \text{s.t.} \quad & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \min_{y} \quad -(y_1(x_1 + 3x_2) + y_2(4x_1 + 2x_2)) \\ & \qquad \text{s.t.} \quad y_1 + y_2 = 1 \\ & \qquad \qquad y_1, y_2 \geq 0 \end{aligned}$$

Solutions:

(a) Global: $x^* = (\frac{1}{4}, \frac{3}{4})$, $y^* = (0, 1)$, $F^* = 1\frac{1}{2}$.

5. Problem: BF4 from [12].

$$\begin{aligned} \min_{x,y} \quad & -(x_1(y_1 + 3y_2) + x_2(4y_1 + 2y_2)) \\ \text{s.t.} \quad & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \text{ and } y \text{ solves} \end{aligned}$$

$$\begin{aligned} \text{LLP}(x): \quad \min_{y} \quad & y_1(2x_1 + 3x_2) + y_2(4x_1 + x_2) \\ \text{s.t.} \quad & y_1 + y_2 = 1 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = (\frac{1}{2}, \frac{1}{2})$, $y = (\gamma, 1-\gamma)$ (for $0 \leq \gamma \leq 1$), $F^* = -2\frac{1}{2}$.

6. **Problem: GS1 from [44].**

$$\begin{aligned} \min_{x,y} \quad & x^2 + (y-10)^2 \\ \text{s.t.} \quad & 0 \leq x \leq 15 \\ \text{and} \quad & y \text{ solves} \\ & \text{LLP}(x) \quad \min_{x,y} \quad (x + 2y - 30)^2 \\ & \qquad \text{s.t.} \quad x + y \leq 20 \\ & \qquad \qquad 0 \leq y \leq 20 \end{aligned}$$

Solutions:

(a) **Global:** $x^* = 2$, $y^* = 14$, $F^* = 20$.

7. **Problem: IA1 from [50].**

$$\begin{aligned} \min_{x,y} \quad & 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 \\ \text{s.t.} \quad & x_1 + x_2 + y_1 - 2y_2 \leq 40 \\ & 0 \leq x_1 \leq 50 \\ & 0 \leq x_2 \leq 50 \\ \text{and} \quad & y \text{ solves} \end{aligned}$$

$$\begin{array}{llrcl} \text{LLP}(x): & \min_y & (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2 & & \\ & \text{s.t.} & -10 \leq y_1, y_2 & \leq & 20 \\ & & 2y_1 - x_1 & \leq & 10 \\ & & 2y_2 - x_2 & \leq & 10 \end{array}$$

Solutions:

(a) Global: $x^* = (0,0)$, $y^* = (-10,-10)$, $F^* = 0$.

(b) Global: $x^* = (0,30)$, $y^* = (-10,10)$, $F^* = 0$.

(c) Local: $x^* = (25,30)$, $y^* = (5,10)$, $F^* = 5$.

(d) Local: $x^* = (20,0)$, $y^* = (0,-10)$, $F^* = 10$.

8. **Problem: SG1 from [62].**

$$\begin{array}{ll} \min_{x,y} & (x-1)^2 + 2y_1^2 - 2x \\ \text{s.t.} & x \geq 0 \text{ and } y \text{ solves} \end{array}$$

$$\begin{array}{llrcl} \text{LLP}(x): & \min_{y \geq 0} & (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1 & & \\ & \text{s.t.} & 4x + 5y_1 + 4y_2 & \leq & 12 \\ & & -4x - 5y_1 + 4y_2 & \leq & -4 \\ & & 4x - 4y_1 + 5y_2 & \leq & 4 \\ & & -4x + 4y_1 + 5y_2 & \leq & -4 \\ & & y_1, y_2 & \geq & 0 \end{array}$$

Solutions:

(a) Global: $x^* = 1\frac{8}{9}$, $y^* = (\frac{8}{9}, 0)$, $F^* = -1\frac{11}{27}$.

(b) Local: $x^* = 0$, $y^* = (1,0)$, $F^* = 1$.

9. **Problem: VSJ1 from [67].**

$$\min_{x,y} \quad \frac{1}{2}(x_1 - 0.8)^2 + \frac{1}{2}(x_2 - 0.2)^2 + \frac{1}{2}(y-1)^2$$

$$\begin{aligned} \text{s.t.} \quad & 0 \le x_1, x_2 \le 1 \\ \text{and} \quad & y \text{ solves} \\ & \quad \text{LLP}(x): \min_y \ \tfrac{1}{2}y^2 + y - x_1 y + 2x_2 y \\ & \qquad\qquad \text{s.t.} \quad 0 \le y \le 1 \end{aligned}$$

Solutions:

(a) $x^* = (\frac{4}{5}, \frac{1}{5})$, $y^* = 0$, $F^* = \frac{1}{2}$.

(b) Local: $x^* = (1, 0)$, $y^* = 0$, $F^* = 0.54$.

10. Problem: VSJ2 from [67].

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}(x_2 - 0.4)^2 + \frac{1}{2}(y - 0.8)^2 \\ \text{s.t.} \quad & 0 \le x_1, x_2 \le 1 \\ \text{and} \quad & y \text{ solves} \\ & \quad \text{LLP}(x): \min_y \ \tfrac{1}{2}y^2 + y - x_1 y + 3x_2 y \\ & \qquad\qquad \text{s.t.} \quad 0 \le y \le 1 \end{aligned}$$

Solutions:

(a) Global: $x^* = (1, \frac{2}{5})$, $y^* = 0$, $F^* = \frac{8}{25}$.

(b) Local: $x^* = (1, 0)$, $y^* = 0$, $F^* = \frac{2}{5}$.

11. Problem: YZ1 from [69].

$$\begin{aligned} \min_{x,y} \quad & x + y \\ \text{s.t.} \quad & -1 \le x \le 1 \\ \text{and} \quad & y \text{ solves} \\ & \quad \text{LLP}(x): \min_y \quad y^2 - 2xy \\ & \qquad\qquad \text{s.t.} \quad -1 \le y \le 1 \end{aligned}$$

Solutions:

(a) Global: $x^* = -1$, $y^* = -1$, $F^* = -2$.

12. Problem: YZZ2 from [70].

$$\begin{array}{lll} \min\limits_{x,y} & x^2 - 2y & \\ \text{s.t.} & x \geq 0 \text{ and } y \text{ solves} & \\ & \text{LLP}(x): \min\limits_{y} & y^2 - 2xy \\ & \qquad\quad \text{s.t.} & 2x - y \leq 0 \\ & & y \geq 0 \end{array}$$

Solutions:

(a) Global: $x^* = 2$, $y^* = 4$, $F^* = -4$.

(b) Local: $x^* = 0$, $y^* = 0$, $F^* = 0$.

A.3 Generated Problems

Because of the numerous global and local solutions to these problems, their solutions are not stated here. Rather, the reader is referred to [28] for details.

A.3.1 Untransformed Problems

$$\begin{array}{lll} \min\limits_{x,y} & \frac{1}{2}\left(\sum\limits_{i=1}^{nx}(x_i - 1)^2 + \sum\limits_{i=1}^{ny} y_i^2\right) & \\ \text{s.t.} & y \text{ solves} & \\ & \text{LLP}(x): \min\limits_{y} \; \sum_{i=1}^{m}(\frac{1}{2}y_i^2 - y_i x_i) & \\ & \qquad\quad \text{s.t.} \; x_i - y_i \leq 1, & i = 1, \cdots, m \\ & \qquad\qquad\; 1 \leq x_i + y_i \leq \rho_i, & i = 1, \cdots, m, \end{array}$$

where

$$
\begin{aligned}
nx &= \text{the number of upper level variables} \\
ny &= \text{the number of lower level variables} \\
m &= \min(nx, ny) \\
\rho &\in \mathbb{R}^m,
\end{aligned}
$$

and $\rho_i \geq 1$ for $i = 1, \cdots, m$.

Each combination of values for nx, ny and ρ defines a different bilevel problem. The following values define the test problems.

1. U1: $nx = 5$, $ny = 5$, and $\rho = [1, 1, \frac{7}{4}, 2, 3]$.
2. U2: $nx = 5$, $ny = 10$, and $\rho = [1\frac{1}{3}, 1\frac{2}{3}, 2, 3\frac{1}{3}, 5\frac{1}{3}]$.
3. U3: $nx = 10$, $ny = 10$, and $\rho = [ones(7), 2, 2, 7]$.
4. U4: $nx = 15$, $ny = 10$, and $\rho = [ones(4), 1\frac{6}{7}, 2 \times ones(3), 5, 9]$.
5. U5: $nx = 15$, $ny = 20$, and $\rho = [ones(3), 1\frac{1}{2} \times ones(6), 2 \times ones(4), 3ones(2)]$.
6. U7: $nx = 50$, $ny = 10$, and $\rho = \frac{1}{5}[5 \times ones(4), 7, 8, 9, 13, 18, 23]$.
7. U8: $nx = 10$, $ny = 60$, and $\rho = [8/5 \times ones(5), 3 \times ones(5)]$.

A.3.2 Transformed Problems

The untransformed problem can also be written in matrix form,

$$
\begin{array}{ll}
\min\limits_{x,y} & \frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} + c^T \begin{bmatrix} x \\ y \end{bmatrix} + \frac{nx}{2} \\
\text{s.t.} & y \text{ solves}
\end{array}
$$

$$\text{LLP}(x): \quad \min_{y} \quad \tfrac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^T S \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\text{s.t.} \quad A\begin{bmatrix} x \\ y \end{bmatrix} \leq b$$

where

$$c = \begin{bmatrix} -ones(nx) \\ zeros(ny) \end{bmatrix} \in \mathbb{R}^n, \qquad S = \begin{bmatrix} zeros(nx, nx) & S_{xy} \\ S_{xy}^T & S_{yy} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$S_{xy} \in \mathbb{R}^{nx \times ny} \text{ satisfies } (S_{xy})_{ij} = \begin{cases} -1 & \text{if } 1 \leq i = j \leq m \\ 0 & \text{otherwise,} \end{cases}$$

$$S_{yy} \in \mathbb{R}^{ny \times ny} \text{ satisfies } (S_{yy})_{ij} = \begin{cases} 1 & \text{if } 1 \leq i = j \leq m \\ 0 & \text{otherwise,} \end{cases}$$

$$b = \begin{bmatrix} ones(m) \\ \rho \\ -ones(m) \end{bmatrix} \in \mathbb{R}^{3m}, \qquad A = \begin{bmatrix} P_x & -P_y \\ P_x & P_y \\ -P_x & -P_y \end{bmatrix} \in \mathbb{R}^{3m \times 3(nx+ny)},$$

$$P_x \in \mathbb{R}^{m \times nx} \text{ and } (P_x)_{ij} = \begin{cases} 1 & \text{if } 1 \leq i = j \leq m \\ 0 & \text{otherwise,} \end{cases}$$

$$P_y \in \mathbb{R}^{m \times ny} \text{ and } (P_y)_{ij} = \begin{cases} 1 & \text{if } 1 \leq i = j \leq m \\ 0 & \text{otherwise.} \end{cases}$$

Using a matrix transformation, this separable quadratic bilevel problem can be transformed into a nonseparable quadratic problem.

Let $v_x \in \mathbb{R}^{nx}$ and $v_y \in \mathbb{R}^{ny}$ be any two vectors satisfying $v_x^T v_x = 1$ and $v_y^T v_y = 1$. Define Householder matrices H_x and H_y using these vectors,

$$\begin{aligned} H_x &= I_{nx} - 2v_x v_x^T \\ H_y &= I_{ny} - 2v_y v_y^T \end{aligned}$$

and let

$$H = \begin{bmatrix} H_x & zeros(nx, ny) \\ zeros(ny, nx) & H_y \end{bmatrix}.$$

Also, let D be a positive definite diagonal matrix, and define $M = HDH$.

For convenience, we shall denote the variables in the transformed problem as t_x and t_y, where

$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} = M^{-1} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Therefore, we can write the transformed problem as

$$\begin{array}{ll} \min\limits_{t_x, t_y} & \frac{1}{2} \begin{bmatrix} t_x \\ t_y \end{bmatrix}^T MM \begin{bmatrix} t_x \\ t_y \end{bmatrix} + c^T M \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \frac{nx}{2} \\ \text{s.t.} & t_y \text{ solves} \end{array}$$

$$\begin{array}{ll} \text{LLP}(t_x): \min\limits_{t_y} & \frac{1}{2} \begin{bmatrix} t_x \\ t_y \end{bmatrix}^T MSM \begin{bmatrix} t_x \\ t_y \end{bmatrix} \\ \text{s.t.} & AM \begin{bmatrix} t_x \\ t_y \end{bmatrix} \leq b. \end{array}$$

Different choices of nx, ny, ρ, v_x, v_y, and D define distinct nonseparable quadratic bilevel problems. The five test problems are defined with the following values. In all cases, let

$$v_x = \frac{w_x}{||w_x||} \quad \text{and} \quad v_y = \frac{w_y}{||w_y||},$$

where w_x and w_y are given below.

1. T1: $nx = 4$, $ny = 2$, $\rho = [1\frac{1}{2}, 3]$, $diag(D) = 10 \times ones(n)$, and

$$\begin{aligned} w_x &= [0.9, 0.4, 0.4, 0.1] \\ w_y &= [0.8, 0.6]. \end{aligned}$$

2. T2: $nx = 5$, $ny = 5$, $\rho = [1, 1\frac{1}{4}, 1\frac{3}{4}, 2, 3]$, $diag(D) = 10 \times ones(n)$, and

$$\begin{aligned} w_x &= \frac{1}{10}[9, 0, 3, 3, 1] \\ w_y &= \frac{1}{10}[0, 8, 6, 0]. \end{aligned}$$

3. T3: $nx = 5$, $ny = 10$, $\rho = [1, 1, 1\frac{4}{5}, 2, 2]$, $diag(D) = ones(n)$, and

$$\begin{aligned} w_x &= [1, 2, -3, 0, 0] \\ w_y &= [2, 1, 0, -2, -3, 0, 0, 1, -1, 4]. \end{aligned}$$

4. T4: $nx = 10$, $ny = 5$, $\rho = [ones(4), 2]$, $diag(D) = ones(n, 1)$, and

$$\begin{aligned} w_x &= [-5, 4, -3, -2, -1, 0, 1, 2, 3, 4] \\ w_y &= [0, 1, 0, -1, 0]. \end{aligned}$$

5. T5: $nx = 10$, $ny = 15$, $\rho = [1, 1, 1, \frac{6}{5}, \frac{7}{5}, \frac{8}{5}, \frac{9}{5}, 3, 5, 7]$, $diag(D) = 10 \times ones(n)$, and

$$\begin{aligned} w_x &= ones(nx) \\ w_y &= [1, 0, 0, 0, 1, 0, -1, -1, -1, 0, 0, 0, 2, 0, 0]. \end{aligned}$$

A.4 Nonlinear Problems

1. Problem: B5 from [10].

$$\begin{aligned} \min_{x,y} \quad & -2x_1^2 - 3x_2 - 4y_1 + y_2^2 \\ \text{s.t.} \quad & x_1^2 + 2x_2 \leq 4 \\ & x \geq 0 \text{ and } y \text{ solves} \\ & \text{LLP}(x): \min_y \; y_1^2 - 5y_2 \\ & \qquad \text{s.t.} \quad x_1^2 - 2x_1 + x_2^2 - 2y_1 + y_2 \geq -3 \\ & \qquad\qquad x_2 + 3y_1 - 4y_2 \geq 4 \\ & \qquad\qquad y \geq 0 \end{aligned}$$

Solutions:

(a) Global: $x^* = (2,0)$, $y^* = (1\frac{3}{5}, \frac{1}{5})$, $F^* = -14.36$.

(b) Local: $x^* = (0,2)$, $y^* = (1\frac{7}{8}, \frac{29}{32})$, $F^* = -12.6787$.

2. **Problem: EB1 from [41].**

$$
\begin{array}{ll}
\min\limits_{x,y} & -x + y_1 + y_2 \\
\text{s.t.} & -1 \le x \le 1 \\
\text{and} & y \text{ solves} \\
& \text{LLP}(x): \min\limits_{y} \; y_2 \\
& \qquad \text{s.t.} \; y_2 - \frac{1}{10} x y_1 \ge 0 \\
& \qquad\qquad 0 \le y_1 \le 1 \\
& \qquad\qquad -1 \le y_2 \le 1
\end{array}
$$

Solutions:

(a) Global: $x^* = 1$, $y^* = (0,0)$, $F^* = -1$.

3. **Problem: YZZ1 from [70].**

$$
\begin{array}{ll}
\min\limits_{x,y} & (x-1)^2 + x^2(y+1) \\
\text{s.t.} & -1 \le x \le 1 \\
\text{and} & y \text{ solves} \\
& \text{LLP}(x): \min\limits_{y} \; y \sin(\frac{\pi x}{2}) \\
& \qquad \text{s.t.} \; -1 \le y \le 1
\end{array}
$$

Solutions:

(a) Global: $x^* = 1$, $y^* = -1$, $F^* = 0$.

(b) Local: $x^* = 0$, $y^* = \alpha$ for $-1 \le \alpha \le 1$, $F^* = 1$.

4. **Problem: Ca Series of Problems - Ca1 ($nx = 1$) and Ca3 ($nx = 3$).**

$$\begin{array}{lll} \min_{x,y} & \sum_{i=1}^{nx} y_i(x_i+1)^2 & \\ \text{s.t.} & -2 \le x_i \le 1, & i = 1:nx \\ \text{and} & y \text{ solves} & \\ & \text{LLP}(x): \min_y \sum_{i=1}^{nx}(y_i^2 - x_i y_i) & \\ & \qquad \text{s.t. } y_i \ge x_i, \quad i = 1:nx & \end{array}$$

Solutions:

(a) **Global:** $x_i^* = -2$, $y_i^* = -1$ for $i = 1:nx$, $F^* = -nx$.

(b) **Local:** ($x_i^* = -\frac{1}{3}$, $y_i^* = -\frac{1}{6}$) or ($x_i^* = -2$, $y_i^* = -1$) for $i = 1:nx$.

5. **Problem: Cb Series of Problems - Cb1 ($nx = 1$) and Cb3 ($nx = 3$).**

$$\begin{array}{lll} \min_{x,y} & \sum_{i=1}^{nx}(x_i^3 + x_i^2 y_i + x_i y_i^2 + y_i^3) & \\ \text{s.t.} & -10 \le x_i \le 10, & i = 1:nx \\ & x_i^2 + x_i y_i + y_i^2 \ge 2, & i = 1:nx \\ \text{and} & y \text{ solves} & \\ & \text{LLP}(x): \min_y \sum_{i=1}^{nx} x_i y_i & \\ & \qquad \text{s.t. } 0 \le x_i y_i^2 + x_i + y_i^2 \le 10, & i = 1:nx. \end{array}$$

Solutions:

(a) **Global:** Unknown.

(b) **Local:** $(x_i, y_i) = (0, \sqrt{2})$ for $i = 1:nx$, $F^* = 2\sqrt{2}nx$.

6. **Problem: Cc Series of Problems - Cc1 ($nx = 1$) and Cc3 ($nx = 3$).**

$$\min_{x,y} \quad \sum_{i=1}^{nx} x_i^2 y_i$$

$$\begin{aligned} \text{s.t.} \quad & -\pi \le x_i \le \pi, \qquad i = 1 : nx \\ & x_i \sin y_i \ge 0, \qquad i = 1 : nx \end{aligned}$$

and y solves

$$\begin{aligned} \text{LLP}(x): \quad \min_{y} \quad & \sum_{i=1}^{nx} x_i y_i^2 \\ \text{s.t.} \quad & -\pi \le y_i \le \pi, \qquad i = 1 : nx \\ & y_i \cos x_i \ge 0, \qquad i = 1 : nx. \end{aligned}$$

Solutions:

(a) Global: $(x_i, y_i) = (-\pi, -\pi)$, $F^* = -\pi^3 nx$.

7. **Problem:** Cd Series of Problems - Cd1 ($nx = 1$) and Cd3 ($nx = 3$).

$$\begin{aligned} \min_{x,y} \quad & \sum_{i=1}^{nx} (x_i^2 y_i^2 - x_i^2 y_i - 2x_i^2 + x_i) \\ \text{s.t.} \quad & x_i^2 + y_i^2 \le 8, \qquad i = 1 : nx \end{aligned}$$

and y solves

$$\begin{aligned} \text{LLP}(x): \quad \min_{y} \quad & \sum_{i=1}^{nx} (y_i - x_i)^2 \\ \text{s.t.} \quad & y_i + 2x_i y_i \ge 0, \qquad i = 1 : nx. \end{aligned}$$

Solutions:

(a) Global: $x_i^* \approx 1.3527$, $y_i^* = x_i^*$, for $i = 1 : nx$, $F^* = -1.4337nx$.

8. **Problem:** Ce Series of Problems - Ce1 ($nx = 1$) and Ce3 ($nx = 3$).

$$\begin{aligned} \min_{x,y} \quad & \sum_{i=1}^{nx} (-x_i^2 y_i - y_i) \\ \text{s.t.} \quad & 9y_i - x_i^2 y_i \ge 0, \qquad i = 1 : nx \end{aligned}$$

and y solves

$$\begin{aligned}
\text{LLP}(x): \quad \min_y \quad & \sum_{i=1}^{nx} x_i y_i^3 \\
\text{s.t.} \quad & y_i \geq 0, & i = 1 : nx \\
& x_i y_i - 2x_i - y_i + 2 \geq 0, & i = 1 : nx.
\end{aligned}$$

Solutions:

(a) Global: $(x_i, y_i) = (-3, 2)$, $F^* = -20nx$.

9. **Problem: Cg Series of Problems - Cg1 ($nx = 1$) and Cg3 ($nx = 3$).**

$$\begin{aligned}
\min_{x,y} \quad & \sum_{i=1}^{nx} (3x_i^4 - 46x_i^3 + 255x_i^2 - 600x_i) \\
\text{s.t.} \quad & 0 \leq x_i(x_i - y_i) \leq 12, \qquad i = 1 : nx \\
\text{and} \quad & y \text{ solves}
\end{aligned}$$

$$\begin{aligned}
\text{LLP}(x): \quad \min_y \quad & \sum_{i=1}^{nx} (3y_i^4 - 20y_i^3 + 12y_i^2 + 96y_i) \\
\text{s.t.} \quad & (x_i - 2)(y_i - 1) \geq 0, & i = 1 : nx \\
& (x_i - 4)(y_i - 3) \geq 0, & i = 1 : nx.
\end{aligned}$$

Solutions:

(a) Global: $x_i^* = \frac{5}{2}$, $y_i^* \in \{1, 3\}$, for $i = 1 : nx$, $F^* = -507\frac{13}{16}nx$.

Appendix B

Notation

Commonly used expressions from the text are listed below, along with the section in which they are defined. Terms which are used only within the section in which they are defined are not included here.

Expression	Section	Meaning
α_1	3.5.1	Step size for which conditions of Lemma 3.5 are satisfied.
α_2	4.2.2	Step size for which conditions of Lemma 4.1 are satisfied.
α_c	4.2.2	Step to first minimum of φ_1 along d_c.
$\alpha_{\mathcal{D}}$	4.2.3	Step to first minimum of φ_2 along $d_{\mathcal{D}}$.
α_{nc}	4.2.4	Step to first minimum of φ_3 along d_{nc}.
α_N	4.2.5	Step to first minimum of φ_4 along d_N.
β_{Ψ}	5.3.2	Algorithm tolerance for a multiplier to be considered in kilter.
β_i	4.2.2	Positive breakpoints of φ_1 along d_c.
$\hat{\beta}_i$	5.2	Positive breakpoints of φ_1 along trust region direction d^k.
$\gamma(w)$	3.5.2	Gradient of differentiable part of p_μ at w.

$\delta(w)$	3.5.1	Differentiable part of p_μ at w
Δ	3.6	Trust region radius.
$\Delta_{\max}$	3.8	Algorithm constant: maximum value of Δ.
ϵ	4.4	Algorithm parameter: activity tolerance.
ζ_j^k	5.2	Generalized Rayleigh Quotient for φ_1^k at $\hat{\beta}_j^k d^k$.
$\eta(w)$	3.5.1	Nondifferentiable part of p_μ at w
$\theta_i(\alpha)$	4.2.2	$\varphi(\alpha d)$ for $\alpha \in (\beta_i, \beta_{i+1})$.
κ	6.6	Multipliers relating gradients of nonbasic and basic activities.
$\kappa_i^n[b_j]$	6.6.2	Multiplier relating nonbasic activity n_i and basic activity b_j.
λ	2.6.4	Lagrange multipliers associated with LLP(x).
Λ	4.2.1	Algorithm parameter: closeness tolerance.
μ	3.3	Penalty parameter.
$\mu_{\max}$	3.4	Algorithm constant: maximum value of μ.
ν_i	4.2.3	Coefficients (one for each dropped activity) used in defining $d_{\mathcal{D}}$.
$\xi(w)$	4.2.5	Vector of values of the activities at w.
$\rho(x)$	3.2	Feasible region of LLP(x).
σ	3.5.2	± 1 value associated with dropping directions in Theorem 3.9.
σ_{nc}	4.2.4	± 1 value used in d_{nc}.
$\tau_{\mathcal{D}}^k$	5.4.1	Maximum measure of curvature at type two point w^k along $d_{\mathcal{D}}^k$.
τ^k	5.2	Maximum measure of curvature at type one point w^k along d^k.
$\tau^{\max}$	5.2	Maximum value of τ^k over all iterations k.
ϕ	3.7	Piecewise quadratic model function of p_μ defined at w.
φ	4.3	Modified version of ϕ actually used in algorithm.
$\varphi_1(w)$	4.2.2	Model function at a type one point for directions in $\mathcal{T}_1$.
$\varphi_2(w)$	4.2.3	Model function at a type two point for directions in $\mathcal{T}_2$.
$\varphi_3(w)$	4.2.4	Model function at a type three point for directions in $\mathcal{T}_3$.

$\chi_a(d_T)$	3.6	Actual decrease in p_μ along d_T from w.
$\chi_p(d_T)$	3.6	Predicted decrease in p_μ along d_T from w.
Ψ	3.5.2	Vector of Lagrange multipliers.
Ψ_i^a	3.5.2	Lagrange multiplier associated with activity a_i.
$\Omega(h, w, \alpha d)$	3.5.2	Generalized Rayleigh Quotient for h from w to $w + \alpha d$.
$\Omega_{p_\mu}(w, \alpha d)$	3.5.2	Measure of second order change in p_μ from w to $w + \alpha d$.
$\Omega_\eta(w, \alpha d)$	3.5.2	Measure of second order change in η from w to $w + \alpha d$.
$\Omega_{\mathcal{D}}^k$	5.4.1	Generalized Rayleigh quotient for φ_2^k along $d_{\mathcal{D}}^k$.
Ω_j^k	5.2	Generalized Rayleigh quotient for φ_1^k at $w + \beta_j^k d_c^k$.
$\mathcal{A}(w)$	3.5.2	Activity matrix at w.
$\mathcal{A}_B$	6.3	Submatrix of $\mathcal{A}$ whose columns form a basis for $\mathcal{A}$.
$\mathcal{A}_N$	6.6.1	Submatrix of $\mathcal{A}$ consisting of the columns not in $\mathcal{A}_B$.
b_1	3.9	Algorithm constant: used to evaluate a successful d_T.
b_2	3.9	Algorithm constant: used to evaluate a very successful d_T.
b_{max}	3.8	Algorithm constant: unboundedness check.
$B(P_=^0)$	6.6.1	Indices in $P_=^0$ for which both $\nabla\lambda_i$ and ∇g_i are in $\mathcal{A}_B$.
$B(S)$	6.6.1	Indices of activities in set S whose gradients are in $\mathcal{A}_B$.
$B(w)$	3.5.2	Hessian of differentiable part of p_μ at w.
$\hat{B}(z)$	3.5.3	Hessian of $p_\mu(w^0 + \alpha d)$ for first order point w^0 and $d = Zz$.
BP	1	Bilevel problem.
BP_C	3.2	Compact form of BP_{KKT}.
BP_{KKT}	2.6.4	One level problem related to BP.
$c_i(x, y, \lambda)$	3.2	Partial derivative with respect to y_i of the Lagrangian function of LLP(x).
d_1	4.2.2	Approximate trust region direction at a type one point.
d_2	4.2.3	Approximate trust region direction at a type two point.
d_3	4.2.4	Approximate trust region direction at a type three point.

d_c	4.2.2	Generalized Cauchy step.
$d_{\mathcal{D}}$	3.5.2	Dropping direction (see also Section 4.2.3).
d_h	4.2.5	Horizontal part of Newton step.
d_{nc}	4.2.4	A direction of negative curvature for H.
d_N	4.2.5	Newton direction.
d_T	3.6	Trust region direction (see also Section 4.2).
d_v	4.2.5	Vertical part of Newton step.
$\mathcal{D}_{P^0_=}$	4.2.3	Indices in $P^0_=$ for which λ_i and g_i are dropped along $d_{\mathcal{D}}$.
$\mathcal{D}^{\lambda}_{P^0_=}$	4.2.3	Indices in $P^0_=$ for which λ_i (but not g_i) is dropped along $d_{\mathcal{D}}$.
$\mathcal{D}^{g}_{P^0_=}$	4.2.3	Indices in $P^0_=$ for which g_i (but not λ_i) is dropped along $d_{\mathcal{D}}$.
$\mathcal{D}_S$	4.2.3	Indices of activities in set S which are dropped along $d_{\mathcal{D}}$.
$f(x, y)$	2.1	Lower level objective function in BP.
$F(x, y)$	2.1	Upper level objective function in BP.
$g(x, y)$	2.1	Lower level constraint functions in BP.
$G(x, y)$	2.1	Upper level constraint functions in BP.
H	3.5.2	Hessian of Lagrangian function at w.
GBP	2.3	Generalized Bilevel Problem
$\mathcal{I}$	2.1	Induced (feasible) region of BP.
$\mathcal{J}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
$\mathcal{K}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
$\mathcal{L}(x, y, \lambda)$	2.6.4	Lagrangian function of LLP(x).
LBP	2.6.1	Linear bilevel problem.
LLP(x)	2.1	Lower level problem in BP, parameterized by x.
m	2.1	Number of lower level variables in BP.
M	2.6.4	$\{1, \cdots, m\}$.
$M'(\epsilon)$	4.4	$M \setminus M^0(\epsilon)$.

$M'(w)$	3.5	$M \setminus M^0(w)$.
$M^0(\epsilon)$	4.4	$\{i \in M : c_i(w) \stackrel{\epsilon}{=} 0\}$.
$M^0(w)$	3.5	$\{i \in M : c_i(w) = 0\}$.
n	2.1	Number of upper level variables in BP.
$N(P^0_=)$	6.6.2	Indices in $P^0_=$ for which only $\nabla\lambda_i$ is in $\mathcal{A}_B$.
$N(S)$	6.6.2	Indices of activities in set S whose gradients are in $\mathcal{A}_N$.
$\mathcal{N}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
p	2.1	Number of lower level constraints in BP.
p_μ	3.3	ℓ_1 penalty function for BP_C.
P	2.1	$\{1, \cdots, p\}$.
$P_=(\epsilon)$	4.4	$\{i \in P : \lambda_i(w) \stackrel{\epsilon}{=} g_i(w)\}$.
$P_=(w)$	3.5	$\{i \in P : \lambda_i(w) = g_i(w)\}$.
$P'_=(\epsilon)$	4.4	$P_=(\epsilon) \setminus P^0_=(\epsilon)$.
$P'_=(w)$	3.5	$P_=(w) \setminus P^0_=(w)$.
$P^+_=(w)$	3.5.2	$\{i \in P'_=(w) : \lambda_i(w) > 0\}$.
$P^-_=(w)$	3.5.2	$\{i \in P'_=(w) : \lambda_i(w) < 0\}$.
$P^0_=(\epsilon)$	4.4	$\{i \in P_=(\epsilon) : \lambda_i(w) \stackrel{\epsilon}{=} 0 \text{ or } g_i(w) \stackrel{\epsilon}{=} 0\}$.
$P^0_=(w)$	3.5	$\{i \in P : \lambda_i(w) = g_i(w) = 0\}$.
$P_\lambda(\epsilon)$	4.4	$\{i \in P \setminus P_=(\epsilon) : \lambda_i(w) < 0\}$.
$P_\lambda(w)$	3.5	$\{i \in P : \lambda_i(w) < g_i(w)\}$.
$P'_\lambda(\epsilon)$	4.4	$P_\lambda(\epsilon) \setminus P^0_\lambda(\epsilon)$.
$P'_\lambda(w)$	3.5	$\{i \in P_\lambda(w) : \lambda_i(w) \neq 0\}$.
$P^0_\lambda(\epsilon)$	4.4	$\{i \in P_\lambda(\epsilon) : \lambda_i \stackrel{\epsilon}{=} 0\}$.
$P^0_\lambda(w)$	3.5	$\{i \in P_\lambda(w) : \lambda_i(w) = 0\}$.
$P_g(\epsilon)$	4.4	$\{i \in P \setminus P_=(\epsilon) : g_i(w) < 0\}$.
$P_g(w)$	3.5	$\{i \in P : g_i(w) < \lambda_i(w)\}$.

$P'_g(\epsilon)$	4.4	$P_g(\epsilon) \setminus P^0_g(\epsilon)$.
$P'_g(w)$	3.5	$\{i \in P_g(w) : g_i(w) \neq 0\}$.
$P^0_g(\epsilon)$	4.4	$\{i \in P_g(\epsilon) : g_i \stackrel{\epsilon}{=} 0\}$.
$P^0_g(w)$	3.5	$\{i \in P_g(w) : g_i(w) = 0\}$.
$\mathrm{PF}(\mu)$	3.3	Penalty function subproblem.
q	4.2.2	Number of variables in BP_C.
$\mathcal{Q}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
r_1	4.3	Algorithm parameter: sufficient decrease coefficient.
$R(x)$	2.1	Rational reaction set of x.
$\mathcal{R}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
t	2.1	Number of upper level constraints in BP.
T	2.1	$\{1, \cdots, t\}$.
$T'(\epsilon)$	4.4	$T \setminus T^0(\epsilon)$.
$T'(w)$	3.5	$\{i \in T : G_i(w) \neq 0\}$.
$T^0(\epsilon)$	4.4	$\{i \in T : G_i(w) \stackrel{\epsilon}{=} 0\}$.
$T^0(w)$	3.5	$\{i \in T : G_i(w) = 0\}$.
$\mathrm{TR}(w, \Delta)$	3.6	Trust region subproblem at w with radius Δ.
u^a_j	6.6.5	Value used in developing necessary optimality conditions.
$\mathcal{U}^*[j]$	6.6.5	Intervals used in developing necessary optimality conditions.
v^a_j	6.6.5	Value used in developing necessary optimality conditions.
$\mathcal{V}^*[j]$	6.6.5	Intervals used in generalized necessary optimality conditions.
w	3.5	$(x, y, \lambda) \in \mathbb{R}^{n+m+p}$.
$\mathcal{W}_1(w)$	4.2.2	Directions which maintain all the activities within the model.
$\mathcal{W}_2(w)$	4.2.3	Directions which maintain the desired activities within the model.
$\mathcal{W}_3(w)$	4.2.4	Directions which maintain all the activities within the model.
$\mathcal{W}_4(w)$	4.2.5	Directions which maintain all the activities within the model.

z_c	4.2.2	The steepest descent direction (in the reduced space).
z_{nc}	4.2.4	Direction of negative curvature for $Z^T H Z$ at a type three point.
$Z(w)$	3.5.2	Orthogonal matrix which satisfies $Z^T A = 0$ at w.
$Z^T H Z$	3.5.2	Reduced Hessian of the Lagrangian function.

Bibliography

[1] E. Aiyoshi and K. Shimizu. Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Transactions on Systems, Man, and Cybernetics*, 11:444–449, 1981.

[2] E. Aiyoshi and K. Shimizu. A solution method for the static Stackelberg problem via penalty method. *IEEE Transactions on Automatic Control*, 29:1111–1114, 1984.

[3] F. Al-Khayyal, R. Horst, and P. Pardalos. Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. *Annals of Operations Research*, 34:125–147, 1992.

[4] N. Alexandrov and J. Dennis. Algorithms for bilevel optimization. Research Report CRPC-TR94474, Center for Research on Parallel Computation, Rice University, 1994.

[5] N. Alexandrov and J. Dennis. Multilevel algorithms for nonlinear optimization. ICASE Report 94-53, Institute for Computer Application in Science and Engineering, NASA, 1994.

[6] G. Anandalingam and D. White. A solution method for the linear static Stackelberg problem using penalty functions. *IEEE Transactions on Automatic Control*, 35:1170–1173, 1990.

[7] G. Anandalingham and T. Friesz. Hierarchical optimization: An introduction. *Annals of Operations Research*, 34:1–11, 1992.

[8] J. Bard. A grid search algorithm for the linear bilevel programming problem. In *Proceedings of the 14^{th} Annual Meeting of the American Institute for Decision Science*, volume 2, pages 256–258, 1982.

[9] J. Bard. Regulating nonnuclear industrial waste by hazard classification. *Journal of Environmental Systems*, 13:21–41, 1983/84.

[10] J. Bard. Optimality conditions for the bilevel programming problem. *Naval Research Logistics Quarterly*, 31:13–26, 1984.

[11] J. Bard. Convex two-level optimization. *Mathematical Programming*, 40:15–27, 1988.

[12] J. Bard and J. Falk. An explicit solution to the multi-level programming problem. *Computers & Operations Research*, 9:77–100, 1982.

[13] J. Bard and J. Moore. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, 11:281–292, 1990.

[14] R. Bartels, A. Conn, and J. Sinclair. Minimization techniques for piecewise differentiable functions: The ℓ_1 solution to an overdetermined linear system. *SIAM Journal on Numerical Analysis*, 15:224–241, 1978.

[15] 0. Ben-Ayed and C. Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38:556–560, 1990.

[16] O. Ben-Ayed, D. Boyce, and C. Blair. A general bilevel linear programming formulation of the network design problem. *Transportation Research B*, 22B:311–318, 1988.

[17] Z. Bi. *Numerical Methods for Bilevel Programming Problems.* PhD thesis, Department of Systems Design Engineering, University of Waterloo, 1992.

[18] Z. Bi, P. Calamai, and A. Conn. An exact penalty function approach for the linear bilevel programming problem. Department of Systems Design Engineering, University of Waterloo.

[19] Z. Bi, P. Calamai, and A. Conn. An exact penalty function approach for the linear bilevel programming problem. Technical Report #167-O-310789, Department of Systems Design, University of Waterloo, 1989.

[20] Z. Bi, P. Calamai, and A. Conn. An exact penalty function approach for the nonlinear bilevel programming problem. Technical Report #180-O-170591, Department of Systems Design, University of Waterloo, 1991.

[21] W. Bialas and M. Karwan. On two-level optimization. *IEEE Transactions on Automatic Control*, AC-27:211–214, 1982.

[22] W. Bialas and M. Karwan. Two-level linear programming. *Management Sciences*, 30:1004–1020, 1984.

[23] J. Bracken, J. Falk, and J. McGill. The equivalence of two mathematical programs with optimization problems in the constraints. *Operations Research*, 22:1102–1104, 1974.

[24] J. Bracken and J. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21:37–44, 1973.

[25] J. Bracken and J. McGill. Defense applications of mathematical programs with optimization problems in the constraints. *Operations Research*, 22:1086–1096, 1974.

[26] J. Bracken and J. McGill. A method for solving mathematical programs with nonlinear programs in the constraints. *Operations Research*, 22:1097–1101, 1974.

[27] R. Burton and B. Obel. The multilevel approach to organizational issues of the firm - a critical review. *Management Science*, 5:395–414, 1977.

[28] P. Calamai and L. Vicente. Generating quadratic bilevel programming problems. *ACM Transactions on Mathematical Software*, 20:103–119, 1994.

[29] W. Candler, J. Fortuny-Amat, and B. McCarl. The potential role of multilevel programming in agricultural economics. *American Journal of Agricultural Economics*, 63:521–531, 1981.

[30] W. Candler and R. Norton. Multi-level programming and development policy. Staff Working Paper 258, World Bank, 1977.

[31] W. Candler and R. Townsley. A linear two-level programming problem. *Computers & Operations Research*, 9:59–76, 1982.

[32] Y. Chen and M. Florian. The nonlinear bilevel programming problem: a general formulation and optimality conditions. Technical Report CRT-794, Centre de Recherche sur les Transports, 1991.

[33] T. Coleman and A. Conn. Second order conditions for an exact penalty function. *Mathematical Programming*, 19:178–185, 1980.

[34] T. Coleman and A. Conn. Nonlinear programming via an exact penalty function: Global analysis. *Mathematical Programming*, 24:137–161, 1982.

[35] T.F. Coleman and A.R. Conn. Second order conditions for an exact penalty function and applications. Research Report CORR 78-27, Department of Combinatorics and Optimization, University of Waterloo, 1978.

[36] A. Conn. Constrained optimization using a nondifferentiable penalty function. *SIAM Journal on Numerical Analysis*, 10:760–784, 1973.

[37] A. Conn, N. Gould, A. Sartenaer, and Ph. Toint. Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal of Optimization*, 3:164–221, 1993.

[38] A. Conn, N. Gould, and Ph. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25:433–460, 1988.

[39] S. Dempe. A simple algorithm for the linear bilevel programming problem. *Optimization*, 18:373–385, 1987.

[40] J. Dennis, Jr. and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Inc., 1983.

[41] T. Edmunds and J. Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21:83–39, 1991.

[42] J. Falk. A linear max-min problem. *Mathematical Programming*, 5:169–188, 1973.

[43] J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32:783–792, 1981.

[44] J. Gauvin and G. Savard. The steepest descent method for the nonlinear bilevel programming problem. École Polytechnique de Montréal.

[45] A. Geoffrion and W. Hogan. Coordination of two-level organizations with multiple objectives. In A.V. Balafrishnan, editor, *Techniques of Optimization*, pages 455–466. Academic Press, New York and London, 1972.

[46] P. Gill, W. Murray, and M. Wright. *Practical Optimization.* Academic Press Inc. (London) Ltd, 1981.

[47] P. Hansen, B. Jaumard, and G. Savard. A variable elimination algorithm for bilevel linear programming. Research Report # 17-89, Rutcor, 1989.

[48] P. Hansen, B. Jaumard, and G. Savard. New branching and bounded rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13:1194–1217, 1992.

[49] A. Haurie, G. Savard, and D. White. A note on: An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 38:553–555, 1990.

[50] Y. Ishizuka and E. Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34:73–88, 1992.

[51] J. Júdice and Faustino. The solution of the linear bilevel programming problem by using the linear complementarity problem. Departamento de Matemática, Universidade do Coimbra.

[52] C. Kolstad. A review of the literature on bi-level mathematical programming. Technical Report LA-10284-MS, UC-32, Los Alamos National Laboratory, 1985.

[53] C. Kolstad and L. Lasdon. Derivative evaluation and computational experience with large bi-level mathematical programs. BEBR Faculty Working Paper 1266, College of Commerce and Business Administration, University of Illinois at Urbana-Champaign, 1986.

[54] L. LeBlanc and D. Boyce. A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows. *Transportation Research B*, 20B:259–265, 1986.

[55] Z. Luo, J. Pang, and S. Wu. Exact penalty functions for mathematical programs and bilevel programs with analytic constraints. Preliminary Version, February 1993.

[56] P. Marcotte. Network design problem with congestion effects: a case of bilevel programming. *Mathematical Programming*, 34:142–162, 1986.

[57] P. Marcotte and D. Zhu. Exact and inexact penalty methods for the generalized bilevel programming problem. GERAD, June 1993.

[58] M Nicholls. The application of non-linear bi-level programming to the aluminum industry. *Journal of Global Optimization*, 8:245–261, 1996.

[59] G. Papavassilopoulos. Algorithms for static Stackelberg games with linear costs and polyhedra constraints. In *Proceedings of the IEEE Conference on Decision and Control*, pages 647–652, New York, 1982. IEEE.

[60] T. Pietrzykowski. An exact potential method for constrained maxima. *SIAM Journal on Numerical Analysis*, 6:299–304, 1969.

[61] D. Ryan and M. Osborne. On the solution of highly degenerate linear programmes. *Mathematical Programming*, 41:385–392, 1988.

[62] G. Savard and J. Gauvin. The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*, 15:265–272, 1994.

[63] H. Scheel and S. Scholtes. Mathematical problems with equilibrium constraints: stationarity, optimality and sensitivity. Preprint submitted to Elsevier Preprint, July 1996.

[64] S. Scholtes and M. Stöhr. Exact penalization of mathematical problems with equilibrium constraints. Spring, 1996.

[65] H. Tuy, A. Migdalas, and P. Värbrand. A quasiconcave minimization method for solving linear two-level programs. *Journal of Global Optimization*, 4:243–263, 1994.

[66] L. Vicente and P. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5:291–306, 1994.

[67] L. Vicente, G. Savard, and J. Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81:379–399, 1994.

[68] H. von Stackelberg. *The theory of the Market Economy*. Oxford University Press, London, 1952.

[69] J. Ye and D. Zhu. Optimality conditions for bilevel programming problems. GERAD, July 1993.

[70] J. Ye, D. Zhu, and Q. Zhu. Generalized bilevel programming problems. GERAD, August 1993.