PH.D. THESIS

A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm

by Craig T. Lawrence
Advisor: Andre Tits

Ph.D. 98-5

ABSTRACT

Title of Dissertation: A COMPUTATIONALLY EFFICIENT FEASIBLE SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM

Craig Travers Lawrence, Doctor of Philosophy, 1998

Dissertation directed by: Professor André L. Tits
Department of Electrical Engineering

The role of optimization in both engineering analysis and design is continually expanding. As such, faster and more powerful optimization algorithms are in constant demand. In this dissertation, motivated by problems from engineering analysis and design, new Sequential Quadratic Programming (SQP) algorithms generating feasible iterates are described and analyzed. What distinguishes these algorithms from previous feasible SQP algorithms is a dramatic reduction in the amount of computation required to generate a new iterate while still enjoying the same global and fast local convergence properties.

First, a basic algorithm which solves the standard smooth inequality constrained nonlinear programming problem is considered. The main idea involves a simple perturbation of the Quadratic Program (QP) for the standard SQP search direction. The perturbation has the property that a feasible direction is always obtained and fast local convergence is preserved. An extension of the basic algorithm is then proposed which solves the inequality constrained minimax problem. The algorithm exploits the special structure of the problem and is shown to have the same global and local convergence properties as the basic algorithm. Next, the algorithm is extended to efficiently solve problems with very many objective and/or constraint functions. Such problems often arise in engineering design as, e.g., discretized Semi-Infinite Programming (SIP) problems. The key feature of the extension is that only a small subset of the objectives and constraints are used to generate a search direction at each iteration. The result is much smaller QP sub-problems and fewer gradient evaluations.

The algorithms have all been implemented and tested. Preliminary numerical results are very promising. The number of iterations and function evaluations required to converge to a solution are, on average, roughly the same as for a widely available state-of-the-art feasible SQP implementation, whereas the amount of computation required per iteration is much less. The ability of the algorithms to effectively solve real problems from engineering design is demonstrated by considering signal set design problems for optimal detection in the presence of non-Gaussian noise.

A COMPUTATIONALLY EFFICIENT FEASIBLE SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM

by

Craig Travers Lawrence

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1998

Advisory Committee:

Professor André L. Tits, Chairman/Advisor
Professor John Baras
Professor P. S. Krishnaprasad
Professor William Levine
Professor Dianne O'Leary

DEDICATION

To Ara and My Parents

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincerest gratitude to Professor André Tits, my advisor. From the beginning he always made me feel more like a colleague than a student, encouraging me to work on problems that interested me, at the same time offering insightful guidance. He was always available and willing to discuss my work, and the experience and knowledge I have gained working with him has been invaluable.

I would also like to thank Dr. Anthony Kearsley and Dr. Paul Boggs of the National Institute of Standards and Technology (NIST). While at NIST I was always encouraged to pursue problems related to my dissertation research. I learned a great deal and gained valuable experience working with Dr. Kearsley and Dr. Boggs applying optimization algorithms to real-world applications.

None of this ever would have been possible if it wasn't for the support and encouragement of my parents. For this I am truly indebted to them. They always provided an environment which made it easy for me to pursue my interests. Not to mention the fact that they were great role models!

Finally, I would like to thank my wife Ara, who has always been by my side, through thick and thin. Her unwavering patience and support were always a source of encouragement, while her love was a source of inspiration.

The research contained in this dissertation was supported by NSF grant numbers DMI-98-13057, DMI-93-13286 and by the NSF Engineering Research Centers Program NSFD-CDR-88-03012.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Optimization in Engineering Design

Optimization plays a critical role in many aspects of engineering analysis and design. In design, once a structure has been chosen, the problem often reduces to that of choosing an “optimal” set of parameters to minimize appropriate “cost” functions subject to constraints imposed by the model and the design specifications. In engineering analysis, optimization proves to be useful, for example, in the study of worst-case performance for a given system.

In this dissertation, optimization algorithms motivated by problems arising from engineering analysis and design are developed. First, the standard inequality constrained nonlinear programming problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{aligned} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable, is considered. In general, such a framework is too rigid to capture many important design problems, though. A much broader class of problems may be tackled if (P) is generalized to the smooth constrained mini-max problem

$$\begin{array}{ll} \min & F(x) \\ \text{s.t.} & g_j(x) \le 0, \quad j = 1, \ldots, m, \end{array} \tag{M}$$

where

$$F(x) \overset{\Delta}{=} \max\{\ f_j(x) \mid j = 1, \ldots, p\ \},$$

and the functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. The balance of this section, as well as the following section, is devoted to discussing how such problems arise in various generic design methodologies. It should become clear that the mini-max framework does indeed provide more freedom and power for the designer.

The so-called method of inequalities (see, e.g., [40]) is based upon the observation that many design problems are naturally posed as simple feasibility problems, i.e. there are no obvious objective (or cost) functions. Design specifications may typically be written in the form

$$g_i(x) \le \epsilon_i, \quad i = 1, \ldots, m, \tag{1.1}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, and x represents the vector of *design parameters.* An algorithm based on the method of inequalities simply searches for a vector x satisfying (1.1). Varying the parameters ϵ_i, $i = 1, \ldots, m$, allows the designer to explore various trade-offs (see Section 1.2).

As an example, consider a feedback control design problem and suppose x represents the feedback gains. The specifications may require that the closed-loop step response fit within a given envelope (see Figure 1.1, which is borrowed from [76]). For a given set of design parameters x, let $s(x, t)$, $t \in [0, T]$, denote the step response function of the closed-loop system. Define the upper bound

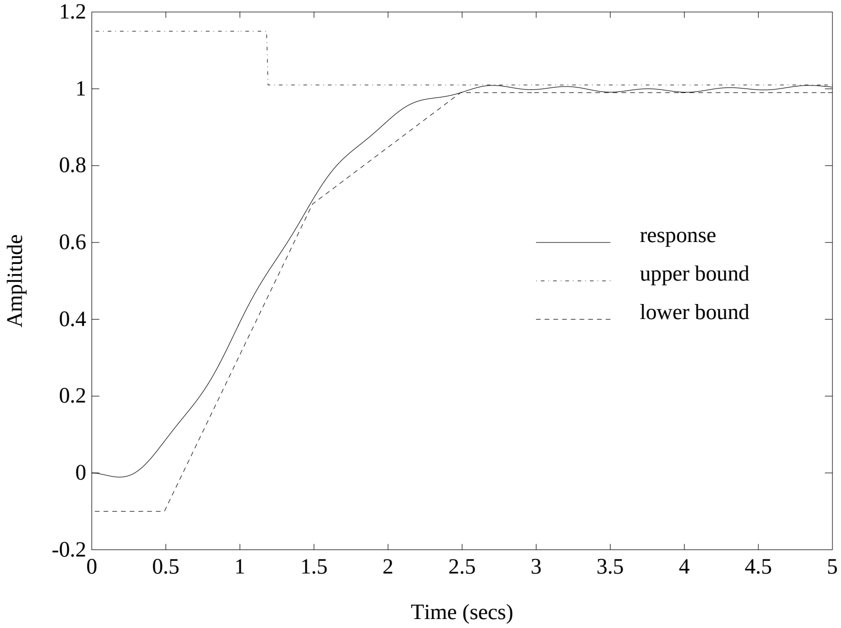

Figure 1.1: Example of a step-response envelope specification.

function as $u(t)$, $t \in [0, T]$, and the lower bound function as $\ell(t)$, $t \in [0, T]$. The envelope specification is then

$$\begin{aligned} s(x,t) - u(t) &\le 0, \quad \forall t \in [0,T], \\ \ell(t) - s(x,t) &\le 0, \quad \forall t \in [0,T]. \end{aligned}$$

Discretizing the time axis into $M + 1$ sample points spaced by $\Delta t = T/M$, the specification is approximated by the set of inequalities

$$\begin{aligned} g_i^u(x) &\stackrel{\Delta}{=} s(x, i \cdot \Delta t) - u(i \cdot \Delta t) \le 0, \quad i = 0, 1, \ldots, M, \\ g_i^\ell(x) &\stackrel{\Delta}{=} \ell(i \cdot \Delta t) - s(x, i \cdot \Delta t) \le 0, \quad i = 0, 1, \ldots, M. \end{aligned}$$

As an aside, in Chapter 5 we will discuss an extension of the algorithms developed in Chapters 3 and 4 to efficiently handle problems with a very large number of constraints and objectives (as in the current example when M is large).

Once a design problem has been translated into a set of inequalities $g_i(x) \le 0$,

$i = 1, \ldots, m$, (here the right-hand side parameters ϵ_i are absorbed into the functions $g_i(\cdot)$) the method of inequalities is reduced to solving

$$\text{find} \ \ x \in \mathbb{R}^n \ \ \text{such that} \ \ g_i(x) \leq 0, \ \ i = 1, \ldots, m.$$

One possible approach for tackling this problem is to consider the unconstrained mini-max problem

$$\min_x \max_i g_i(x),$$

which, under the appropriate regularity assumptions, is an instance of (M). Of course, since we only require a feasible point, it is only necessary to iterate on this mini-max problem until some x^* is found such that

$$\max_i g_i(x^*) \leq 0.$$

Another closely related design methodology is the so-called multi-objective (or multi-criterion) optimization approach (see, e.g., [40]). In this approach, the design problem is translated into a set of performance objectives:

$$\{f_i(x) \mid i = 1, \ldots, p\}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, p$, and x represents a vector of design parameters. The performance objectives are formulated so that, for objective i, design x' is "better" than x'' if

$$f_i(x') < f_i(x'').$$

Of course, in general, multiple competing objectives cannot be simultaneously minimized. Instead, they must be combined into a single composite objective function which is then minimized. A common choice is the weighted max function. Note that it is typically meaningless to directly compare two competing objectives (e.g., it makes no sense to directly compare the value of a stability margin with that of rise time in control system design). Thus, the designer must assign scalings (weights) c_i, $i = 1, \dots, p$, which allow meaningful comparison between the scaled (dimensionless) objective functions

$$\frac{f_i(x)}{c_i}, \quad i = 1, \dots, p.$$

A reasonable choice for the scaling factors is the difference between a value the designer considers "bad" and one that is "good" for the objective (see Section 1.2). Once scaled, the design problem is reduced to the unconstrained mini-max problem

$$\min_x \max_i \frac{f_i(x)}{c_i}.$$

Additional specifications in the form of inequality constraints are often appended to the problem. Such a situation occurs when a quantity is required to be below a given threshold and there is no need to expend additional effort on further reduction. Clearly, in this case, the design problem is reduced (for a fixed set of scaling factors) to solving a constrained mini-max problem of the form (M).

1.2 Interactive Optimization-Based Design

If optimization is to be an effective and useful tool for engineering design, it should allow for, if not enhance, trade-off exploration in an interactive design environment. It is typically impossible for a designer to rigidly specify the various objectives and constraints in advance. Indeed, a more realistic approach allows one to initially specify "approximate" versions of the objectives and constraints. The optimization then proceeds interactively, allowing the designer to tighten or relax specifications as he/she sees fit, depending upon the quality and suitability of intermediate "solutions". Such an approach has been proposed in [44, 72].

An important concept with respect to trade-off analysis is that of *Pareto optimality.* Ignoring constraints for now (the definitions generalize to the constrained case in a straightforward manner), recall the multi-objective optimization-based design discussion from Section 1.1. A "design" x is said to be Pareto optimal if, in a neighborhood of x, a reduction in any one of the objectives f_i can only be achieved at the expense of increasing one of the others. The set of all such x parameterizes the so-called Pareto optimal set (see Figure 1.2 for an example with $p = 2$). It should be clear that if a design is to be considered optimal, it must parameterize a point somewhere in the Pareto optimal set. It is typically not clear, though, which point on the surface is the "best". Trade-off exploration, which amounts to searching the Pareto optimal set, is accomplished by adjusting the scaling factors c_i, $i = 1, \ldots, p$ and solving the resultant optimization

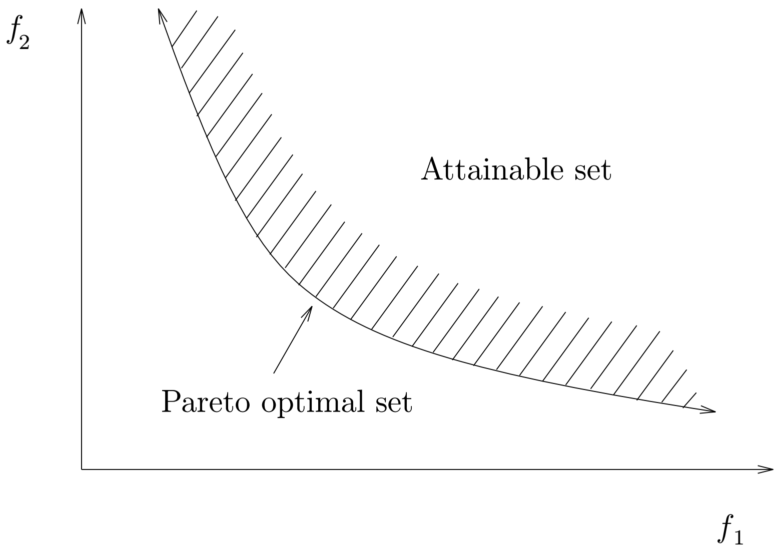

Figure 1.2: Pareto optimality.

problem. Of course this adjustment cannot be done algorithmically as it relies entirely upon qualitative judgments by the designer.

An interactive optimization-based design methodology proposed by Nye and Tits [44] will be briefly described in the remainder of this section. The algorithms developed in this dissertation are ideal candidates for use in such a design approach. Our exposition and notation will closely follow that of [72].

Suppose that a structure has been chosen for the design (e.g., a state feedback controller) and all that remains is to choose design parameter values subject to a set of given specifications. In [44], the next step is for the designer to partition the set of specifications into three classes.

- **Hard Constraints** - Specifications which *must* be satisfied. For example, closed-loop stability or physical realizability.

- **Soft Constraints** - Specifications involving a target value which the design should approach if possible, and requiring no further improvement once the target value is reached. For example, stability under plant uncertainty or controller bandwidth.

- **Objectives** - Specifications which should be minimized or maximized. For example, closed-loop sensitivity to disturbances and plant variations or the integral of the squared error of a step response.

The next step is for the designer to assign *good values* and *bad values* to each of the soft constraints and objectives. These values are assigned according to the so-called *uniform satisfaction/dissatisfaction rule*, i.e.

> "Having any one of the various objectives or soft constraints achieve its *good* value should provide the same level of satisfaction to the

> designer, while having any one of them achieve their *bad* value should provide the same level of dissatisfaction."[72]

Based on these values, the objectives and soft constraints are scaled according to

$$\texttt{scaled\_value} = \frac{\texttt{raw\_value} - \texttt{good\_value}}{\texttt{bad\_value} - \texttt{good\_value}},$$

where `raw_value` is the actual value of the specification. Note that the hard constraints are also assigned good and bad values, but the good value is the only important threshold in this case (the bad value need only be consistent).

Let $\texttt{hard}_j(\cdot)$ denote the scaled hard constraint functions, $\texttt{soft}_i(\cdot)$ denote the scaled soft constraint functions, and $\texttt{obj}_k(\cdot)$ denote the scaled objective functions. The interactive optimization process proceeds in three steps.

- **Phase I** - Attempt to generate a design in which all hard constraints are satisfied. Each iteration decreases the maximum hard constraint violation by working on the problem

$$\min_x \max_j \ \texttt{hard}_j(x).$$

- **Phase II** - Entered when all hard constraints are satisfied, but not all soft constraints and objectives have achieved their good values. Each iteration improves the maximum value among scaled objectives and soft constraints, while maintaining feasibility for hard constraints, by iterating with a feasible direction algorithm on the problem

$$\begin{array}{ll} \min\limits_x & \max\limits_{k,i}\{\texttt{obj}_k(x), \texttt{soft}_i(x)\} \\ \text{s.t.} & \texttt{hard}_j(x) \le 0, \quad \forall j. \end{array}$$

- **Phase III** - All hard constraints are satisfied and all objectives and soft constraints have achieved their good values. Each iteration improves the worst objective, while maintaining feasibility for all constraints, by iterating with a feasible direction algorithm on the problem

$$\begin{aligned} \min_{x} \quad & \max_{k} \ \texttt{obj}_k(x) \\ \text{s.t.} \quad & \texttt{hard}_j(x) \leq 0, \quad \forall j, \\ & \texttt{soft}_i(x) \leq 0, \quad \forall i. \end{aligned}$$

At any point during the optimization, the designer may stop the process and adjust the good/bad values. In particular, it is through adjusting the good and bad values that the designer may search the Pareto optimal set, hence exploring trade-offs in the design. Of course, it may become clear that it is impossible to achieve an acceptable design. In such a case the designer may choose to modify the structure of the design and begin the entire process again.

A number of requirements on the underlying optimization algorithm are imposed by such a methodology. To begin with, the algorithm must be able to solve an inequality constrained mini-max problem (M). Further, it should be clear from the above discussion that the algorithm must generate *feasible* iterates, i.e. iterates which satisfy all inequality constraints (see Section 1.3). Next, the successive iterates should always improve the maximum objective function value, that is

$$\max_{k} \texttt{obj}_k(x_{i+1}) < \max_{k} \texttt{obj}_k(x_i).$$

In addition to these requirements, the algorithm should require as few function evaluations as possible, since evaluating functions is often expensive in an engineering context. Finally, subject to all of the above, the algorithm should be as fast as possible. The algorithms developed in this dissertation are ideal for such an application.

1.3 Feasibility

Denote the feasible set for the problems (M) and (P) by

$$X \overset{\Delta}{=} \{\ x \in \mathbb{R}^n \mid g_j(x) \leq 0,\ j = 1, \ldots, m\ \}.$$

In [53, 25, 48, 51, 3], variations on the standard Sequential Quadratic Programming (SQP) iteration (see Section 2.3) for solving (P) are proposed which generate iterates lying within X. Such methods are sometimes referred to as "Feasible SQP" (or FSQP) algorithms. It was observed that requiring feasible iterates has both algorithmic and application-oriented advantages. Algorithmically, feasible iterates are desirable because

- The Quadratic Programming (QP) subproblems are always consistent, i.e. a feasible solution always exists, and
- The objective function may be used directly as a merit function in the line search.

State of the art SQP algorithms typically include complex schemes to deal with inconsistent QPs. Further, the choice of an appropriate merit function (to enforce global convergence) is not always clear. Requiring feasible iterates eliminates these issues. In an engineering context, feasible iterates are important because

- Often objective functions are undefined outside of the feasible region X,
- The optimization process may be stopped after a few iterations, yielding a feasible point, and

- Trade-offs may be meaningfully explored.

These features are all relevant in both engineering analysis and design. For a situation in which an objective function may be undefined outside of the feasible region, consider control design problems where stability or physical realizability are among the constraints. If the system is unstable, certain specifications on, for example, a time response may be undefined, e.g. settling time. The second point above is critical for real-time applications. In such applications, a feasible point may be required before the algorithm has had time to "converge" to a solution. Finally, the last point is directly related to the discussion of the previous section. To begin with, the interactive design methodology of [44] specifically requires an optimization algorithm generating feasible iterates. In general, though, it doesn't make sense to explore trade-offs by relaxing or tightening certain specifications before all specifications have been satisfied, i.e. are feasible.

1.4 Objective and Contributions

The objective of this dissertation is to develop and analyze computationally efficient feasible SQP algorithms. We begin with a core algorithm, then extend it to handle the mini-max problem, and finally incorporate a scheme for efficiently solving problems with a large number of objectives and/or constraints. The contributions are summarized as follows.

- A new SQP algorithm generating feasible iterates requiring the solution of only one QP and (at most) two linear least squares problems per iteration.
  - The algorithm is shown to be globally convergent.

  - The local convergence rate is shown to be 2-step superlinear.
  - Numerical experiments show it performs very well in practice.

- The algorithm is extended to handle the mini-max problem in a way which exploits the problem structure.
  - It is proved that the global and local convergence properties are preserved.
  - Numerical experiments again show the algorithm performs very well in practice.

- The algorithm is equipped with a scheme to allow it to efficiently solve problems with a very large number of objectives and/or constraints.
  - It is again proved that the global and local convergence properties are preserved.
  - The size of the sub-problems and the number of gradient evaluations are dramatically reduced.
  - Numerical experiments again show the algorithm performs very well in practice.

- A high-quality C implementation of the algorithms.

- Application of the algorithms to a problem from engineering design, specifically the design of optimal signal sets for transmission in the presence of non-Gaussian noise.

1.5 Outline

Broadly, this dissertation is organized as follows. After a brief discussion of some relevant background material, the core algorithm is presented and analyzed. In the two chapters that follow, the algorithm is extended and the appropriate convergence analysis is given for each case. These three chapters constitute the bulk of the contribution (including all of the theoretical contribution) of the work. The focus then changes to implementation issues and an application of the algorithms, followed by concluding remarks. The balance of this section outlines the content in more detail.

In Chapter 2 we review concepts from the theory of nonlinear programming which are directly relevant and important to the material that follows. The topics include optimality conditions (first and second order) for general minimax problems, the notions of global convergence and rates of local convergence, a brief introduction to SQP algorithms, and finally a discussion of algorithms which generate feasible iterates. While not intended to be an exhaustive tutorial (it is assumed the reader is familiar with these concepts), the chapter is meant to serve as a brief review.

Chapter 3 presents the core algorithm and analysis which forms the foundation of the dissertation. The basic idea involves a simple perturbation of the SQP search direction and a technique for iteratively updating the perturbation. Inspiration for the requirements on the perturbation is drawn from a well-known feasible SQP algorithm with strong convergence properties. Under mild assumptions, the algorithm is shown to be globally convergent and locally 2-step superlinearly convergent. In order to show 2-step superlinear convergence we call on a modified version of a well-known argument due to Powell. The modification of the argument is provided in an appendix to the chapter. Several implementation details, as well as promising numerical results, are also discussed.

The core algorithm of Chapter 3 is extended in Chapters 4 and 5. In Chapter 4, the constrained mini-max problem is considered and an algorithm is given which takes advantage of the mini-max structure of the problem. One of the key advantages, among others, over reformulating the problem as a standard constrained nonlinear programming problem (as is often done) is that we maintain the objective function descent property. This is useful in many contexts. In addition, we simplify matters since it is unnecessary to waste effort maintaining "feasibility" for constraints which are actually converted objectives. Problems which have very many objectives and/or constraints, e.g. discretized problems from Semi-Infinite Programming (SIP), are the subject of Chapter 5. In this chapter, the algorithms of Chapters 3 and 4 are equipped with a scheme which greatly reduces the size of the sub-problems at each iteration as well as the number of gradient evaluations. This is accomplished by carefully choosing only a subset of the objectives and constraints in order to construct the search direction at each iteration. In both chapters a complete convergence analysis is given, as well as important implementation details and numerical results.

In Chapter 6 a complete problem statement is given and the structure of the implementation, with calling sequence and description of the input and output parameters, is provided. In addition, we discuss how the implementation deals with an infeasible initial point, how we maintain and update Cholesky factors for the Hessian approximation, a scheme for making the linear algebra more efficient, and an option to allow the use a full QP to compute the Maratos correction. An application to a real problem from engineering design is considered in Chapter 7.

The problem involves the design of signal sets to be transmitted over an additive noise channel in which the noise distribution is not necessarily Gaussian. As there are many local solutions, the algorithms from this dissertation are incorporated into a stochastic global algorithm in an attempt to locate globally optimal signal sets.

Finally, in Chapter 8 we briefly sum up and discuss several directions for future research. Most of the proposed future work involves further extensions of the algorithm to handle a broader class of problems, in addition to improvements in the implementation and more extensive testing and tuning.

Chapter 2

Background

2.1 Optimality Conditions

In this section we briefly review some fundamental concepts from nonlinear programming. Under appropriate assumptions, *optimality conditions* provide a characterization of solutions and, in some cases, suggest methods of finding such solutions. For a more detailed discussion of optimality conditions and their implications see, for example, the texts [39, 1]. For the sake of generality, we will consider only the mini-max problem (which is repeated here for the sake of convenience)

$$\begin{aligned} &\min \quad F(x) \\ &\text{s.t.} \quad g_j(x) \leq 0, \qquad j = 1, \ldots, m, \end{aligned} \tag{M}$$

where

$$F(x) \overset{\Delta}{=} \max\{\ f_j(x) \mid j = 1, \ldots, p\ \},$$

and the functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. A point $x \in \mathbb{R}^n$ is said to be a *Karush-Kuhn-Tucker*

(KKT) point[1] for the problem (M) if there exist scalars (*KKT multipliers*) μ^j, $j = 1, \ldots, p$, and λ^j, $j = 1, \ldots, m$, satisfying

$$\begin{cases} \displaystyle\sum_{j=1}^{p} \mu^j \nabla f_j(x) + \sum_{j=1}^{m} \lambda^j \nabla g_j(x) = 0, \\ \displaystyle\sum_{j=1}^{p} \mu^j = 1, \\ g_j(x) \le 0, \quad j = 1, \ldots, m, \\ \mu^j \left(f_j(x) - F(x)\right) = 0 \text{ and } \mu^j \ge 0, \quad j = 1, \ldots, p, \\ \lambda^j g_j(x) = 0 \text{ and } \lambda^j \ge 0, \quad j = 1, \ldots, m. \end{cases} \tag{2.1}$$

Define the active sets

$$I(x) \overset{\Delta}{=} \{\ j \mid g_j(x) = 0\ \},$$
$$J(x) \overset{\Delta}{=} \{\ j \mid f_j(x) = F(x)\ \}.$$

To see why KKT points are of interest, consider the set of all directions which point strictly into the feasible set at a (feasible) point x, i.e.

$$\mathcal{D}(x) \overset{\Delta}{=} \{\ d \in \mathbb{R}^n \mid \langle \nabla g_j(x), d \rangle < 0, \ \ \forall j \in I(x)\ \}.$$

We assume for this discussion that some form of *constraint qualification* holds at the point x which ensures that $\mathcal{D}(x)$ is not empty. With some thought, it should be clear that if x is a local minimizer for (M), then along each direction in $\mathcal{D}(x)$ at least one active objective function must increase, i.e.

$$\langle \nabla f_j(x), d \rangle \ge 0, \ \ \text{for some } j \in J(x), \ \forall d \in \mathcal{D}(x).$$

It is not difficult to show that this is equivalent to the condition

$$\nexists\ d \in \mathbb{R}^n \ \ \text{such that} \ \ \begin{cases} \langle \nabla f_j(x), d \rangle < 0, \quad j \in J(x), \\ \langle \nabla g_j(x), d \rangle < 0, \quad j \in I(x), \end{cases}$$

[1] These conditions are easily obtained from the more familiar KKT conditions for the case $p = 1$ by considering the equivalent single-objective problem (M') introduced in Section 4.1.

which is, in turn, equivalent to 0 being in the convex hull of the active objective and constraint gradients. The following theorem, which is well-known (see, e.g., Section 10.8 of [39] for the $p = 1$ case), follows from these observations.

Theorem 1. *Suppose that x^* is a local minimizer for (M) and the set*

$$\{\ \nabla g_j(x^*) \mid j \in I(x^*)\ \}$$

is linearly independent. Then x^* is a KKT point for (M).

It follows that, since (2.1) involves only first derivatives, being a KKT point is a first-order *necessary* condition of optimality. When no assumptions concerning convexity are made, in order to obtain a *sufficient* condition for optimality, we will need to appeal to higher order derivatives. This, of course, implies we will have to assume higher order derivatives exist. An important function associated with the problem (M) is the *Lagrangian* $L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ defined as

$$L(x, \mu, \lambda) \triangleq \sum_{j=1}^{p} \mu^j f_j(x) + \sum_{j=1}^{m} \lambda^j g_j(x).$$

Suppose that x^* satisfies the first-order optimality conditions (2.1) with multipliers $\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^m$. Further, suppose that x^* is a *regular* point, i.e. the set $\{\ \nabla g_j(x^*) \mid j \in I(x^*)\ \}$ is linearly independent. Then x^* is said to satisfy the *second order sufficiency conditions* if $\nabla^2_{xx} L(x^*, \mu^*, \lambda^*)$ is positive definite on the subspace

$$\begin{aligned}\{h \mid \langle \nabla f_i(x^*), h\rangle = \langle \nabla f_j(x^*), h\rangle,\ \forall i, j \in J(x^*)&\\ \text{and } \langle \nabla g_j(x^*), h\rangle = 0,\ \forall j \in I(x^*)\}.&\end{aligned}$$

It is said that *strict complementary slackness holds* if we also have $\mu^{*,j} > 0$, for all $j \in J(x^*)$, and $\lambda^{*,j} > 0$ for all $j \in I(x^*)$. The following theorem (again, see [39] for the $p = 1$ case) establishes the sufficiency of these conditions.

Theorem 2. *Suppose that $x^* \in \mathbb{R}^n$ satisfies the second-order sufficiency conditions with strict complementary slackness. Then x^* is a strict local minimizer.*

2.2 Convergence

When analyzing iterative algorithms such as those presented in the following chapters, there are two questions of primary interest concerning the sequences which are generated. First, given an arbitrary initial point, will the sequence converge to some "desirable" point? An algorithm which is guaranteed to generate a sequence converging to a desirable point is said to be *globally* convergent. Once this has been established, attention is turned to the question of how fast the sequence will converge. The answer to this question is commonly referred to as the *local* rate of convergence for the algorithm. In this section we will briefly discuss a few well-known asymptotic convergence rate indicators which are relevant to our discussions. For a comprehensive discussion of rates of convergence for iterative algorithms, see [46].

Here we will be exclusively interested in the so-called *quotient* convergence rates. A sequence $\{x_k\}$ is said to converge to x^* with Q-order $p \geq 1$ and Q-factor γ if there exist $\underline{k}$ such that

$$\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|^p, \quad \forall k \geq \underline{k}.$$

Of particular importance are the special cases

- **Q-linear** : $p = 1$ and $\gamma \in\ (0, 1)$, equivalently

$$\lim_{k\to\infty} \sup \frac{\|x_k - x^*\|}{\|x_{k+1} - x^*\|} < 1,$$

- **Q-superlinear** :

$$\lim_{k\to\infty} \frac{\|x_k - x^*\|}{\|x_{k+1} - x^*\|} = 0,$$

- **Q-quadratic** : $p = 2$, and $\gamma > 0$, equivalently

$$\lim_{k\to\infty} \sup \frac{\|x_k - x^*\|}{\|x_{k+1} - x^*\|^2} < \infty.$$

Of course, Q-quadratic convergence, well known to be the convergence rate for Newton's method of finding roots of nonlinear equations, is the fastest of the three. Note that a sequence which converges Q-superlinearly (which we will refer to simply as superlinear convergence) may not converge with any Q-order $p > 1$. Thus, while being faster than linear convergence, superlinear convergence does not imply quadratic. In general, we will be interested in establishing superlinear convergence, or, specifically, the slightly weaker notion of 2-step superlinear convergence, i.e.

$$\lim_{k\to\infty} \frac{\|x_{k-1} - x^*\|}{\|x_{k+1} - x^*\|} = 0.$$

Quadratic convergence typically comes at the price of requiring higher order derivatives than we are willing to assume available to the algorithm.

2.3 SQP Algorithms

Sequential Quadratic Programming (SQP) has evolved into a broad classification encompassing a variety of algorithms. For the sake of simplicity, we consider the problem (P) in this section, which we repeat for convenience,

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{aligned} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. When the number of variables n is not too large, SQP algorithms are widely acknowledged to be the most successful algorithms available for solving (P). For an excellent recent survey of SQP algorithms, and the theory behind them, see [5].

In general, an SQP algorithm is characterized as one in which a quadratic model of (P) is formed at the current estimate of the solution and is solved in order to construct the next estimate of the solution. Typically, in order to ensure global convergence, a suitable *merit function* is used to perform a line search in the direction provided by the solution of the quadratic model. While such algorithms are potentially very fast, the local rate of convergence is critically dependent upon the type of second order information utilized in the quadratic model as well as the method by which this information is updated.

Given estimates $x_k \in \mathbb{R}^n$ of the solution of (P), $0 \leq \lambda_k \in \mathbb{R}^m$ of the Lagrange multipliers at the solution, and $0 < H_k = H_k^T \in \mathbb{R}^{n \times n}$ of the Hessian of the Lagrangian $L(x_k, \lambda_k)$, the standard[2] SQP search direction $d_k^0 = d^0(x_k, H_k) \in \mathbb{R}^n$ is computed as a solution of the QP

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle d^0, H_k d^0 \rangle + \langle \nabla f(x_k), d^0 \rangle \\ \text{s.t.} \quad & g_j(x_k) + \langle \nabla g_j(x_k), d^0 \rangle \leq 0, \quad j = 1, \ldots, m. \end{aligned} \tag{$QP^0(x_k, H_k)$}$$

[2] This is *not* the only choice available for an SQP search direction, though it is the most popular.

Most SQP algorithms require that the estimates of the Lagrange multipliers be updated as well. Let $\lambda_k^0 \in \mathbb{R}^m$ be the optimal multipliers from $QP^0(x_k, H_k)$. One possible choice for a search direction in the multiplier space is $d_k^\lambda = \lambda_k^0 - \lambda_k$. Thus, for a suitable step-length parameter $t_k \in (0, 1]$, new estimates of the solution of (P) and the corresponding multipliers may be taken as

$$x_{k+1} = x_k + t_k d_k^0, \quad \lambda_{k+1} = \lambda_k + t_k d_k^\lambda.$$

Another popular alternative for the multiplier update is to simply set $\lambda_{k+1} = \lambda_k^0$. While not all SQP algorithm follow precisely this, a basic framework is as follows.

Algorithm SQP

```
Data: $x_0 \in \mathbb{R}^n$, $0 < H_0 = H_0^T \in \mathbb{R}^{n \times n}$, and a merit function $\phi(\cdot)$.

Step 0 - Initialization. set $k \leftarrow 0$.

Step 1 - Computation of search direction. compute $d_k^0 = d^0(x_k, H_k)$
   and the corresponding QP multiplier vector $\lambda_k^0$.

Step 2 - Line search. compute $t_k$ such that

   $\phi(x_k + t_k d_k^0) < \phi(x_k)$.

Step 3 - Updates.

   (i). set $x_{k+1} \leftarrow x_k + t_k d_k^0$ and $\lambda_{k+1} \leftarrow \lambda_k + t_k d_k^\lambda$.

   (ii). compute a new estimate $H_{k+1}$ of the Hessian of the Lagrangian.

Step 4. if convergence criterion is satisfied, then stop.
   else set $k \leftarrow k + 1$ and goto Step 1.
```

Clearly there are a number of “degrees of freedom” that must be fixed before such an algorithm could be implemented. In terms of global and local convergence properties, the two most important choices to be made are that of an appropriate merit function $\phi(\cdot)$ and the Hessian updating scheme to be used in *Step 3(ii)*. The purpose of the merit function is to enforce global convergence far from the solution. The requirements of decreasing the objective function and satisfying the constraints must be balanced. In order to measure progress towards a solution, a merit function is typically chosen so that its unconstrained minimizers correspond to minimizers of (P). In order for the algorithm to be well-posed, it is necessary that the computed search direction d_k^0 is a descent direction at x_k for the merit function, i.e. there must exist a $\bar{t} > 0$ such that

$$\phi(x_k + td_k^0) < \phi(x_k), \quad \forall t \in (0, \bar{t}\,].$$

A common example is the ℓ_1 merit function

$$\phi_{\ell_1}(x) = f(x) + \rho \cdot \sum_{i=1}^{m} g_i^+(x),$$

originally proposed by Han [22], where $\rho > 0$ and $g_i^+(x) = \max\{0, g_i(x)\}$. If ρ is chosen large enough, the unconstrained minimizer of $\phi_{\ell_1}(\cdot)$ is, in fact, a solution of (P).

An obvious choice for the matrices H_k is, of course, the exact Hessian of the Lagrangian evaluated at (x_k, λ_k), i.e.

$$H_k = \nabla^2_{xx} L(x, \lambda)\big|_{(x_k,\lambda_k)}.$$

If (x_k, λ_k) are sufficiently close to a strong local minimizer (x^*, λ^*), then it can be shown that such a choice leads to a quadratic rate of convergence [5] (assuming a unit step is always accepted in the line search). Unfortunately, in most applications, the computation of second derivatives is prohibitively expensive. Further, the true Hessian is often not well-behaved (not positive definite) far from the solution. Thus, approximate updating schemes are used in most practical algorithms. While a great number of such schemes have been studied in the literature, one particular method that has enjoyed great success in practice is the class of *secant* approximations. Following [5], a Taylor expansion in x of $\nabla_x L(x, \lambda_{k+1})$ about the point x_{k+1} reveals

$$\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1}) \approx \nabla^2_{xx} L(x_{k+1}, \lambda_{k+1})(x_{k+1} - x_k).$$

This relationship inspires the secant equation, which requires an update H_{k+1} to satisfy

$$H_{k+1}(x_{k+1} - x_k) = \nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1}).$$

The most common updating schemes add either a rank-one or rank-two matrix $U_k = \mathcal{U}(H_k, x_{k+1}, x_k, \lambda_{k+1}, \lambda_k)$ to H_k so that $H_{k+1} = H_k + U_k$ will satisfy the secant equation. Under appropriate conditions, such updating schemes lead to superlinear rates of convergence. Finally, we note that simply using a positive definite matrix for all k (such as the identity matrix) without attempting to estimate any second order information will likely result in a linear convergence rate.

Local convergence analysis is always done under the assumption that a full step of one, i.e. $t_k = 1$ is accepted in the line search for all k sufficiently large. It turns out, though, that for certain popular choices of merit functions (e.g., the ℓ_1 merit function) the step length may be truncated even in a neighborhood of the solution, hence preventing superlinear convergence. This phenomenon was first observed by N. Maratos in his PhD thesis [41]. Several methods have been proposed in the literature to overcome this problem. Among these are the "watch-dog" technique of Chamberlain, et al. [9], the non-monotone line search schemes originally proposed by Grippo, et al. [20], then adapted to the SQP framework as a Maratos avoidance scheme by Panier and Tits [50] and Bonnans, et al. [7], and the second-order correction, or "bending", method proposed by Mayne and Polak in [42]. The algorithms discussed in this dissertation will utilize a second-order correction inspired by that in [42].

2.4 Feasible Direction Algorithms

A *feasible direction* at a point $x \in X$ (recall the definition of X from Section 1.3) is defined as any vector $d \in \mathbb{R}^n$ satisfying $x + td \in X$ for all $t \in [0, \bar{t}\,]$, for some $\bar{t} > 0$. Note that the SQP direction $d^0 = d^0(x, H)$, a direction of descent for f, may not be a feasible direction at x, though it is at worst tangent to the active constraint surface (see Figure 2.1, where d is a feasible descent direction and the dashed lines represent level curves of f). Thus, in order to generate feasible iterates in the SQP framework, it is necessary to "tilt" d^0 into the feasible set. A number of different approaches have been considered in the literature for generating feasible directions and, specifically, tilting the SQP direction.

Early feasible direction algorithms (see, e.g., [80, 53]) were first-order methods, i.e. only first derivatives were used and no attempt was made to accumulate and use second-order information. Furthermore, search directions were often computed via linear programs instead of QPs. As a consequence, such algorithms converged linearly at best. Polak proposed several extensions to these algorithms (see [53], Section 4.4) which took second-order information into account when

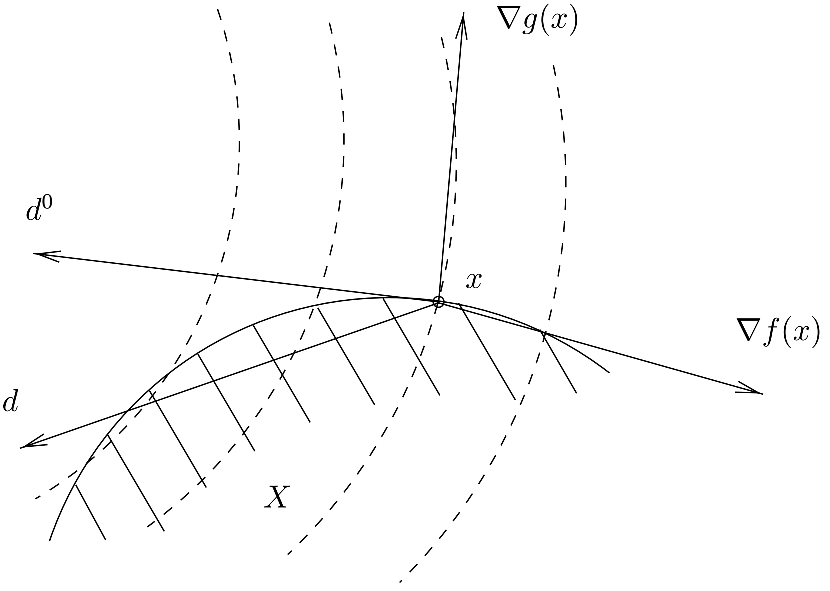

Figure 2.1: Infeasibility of SQP direction d^0.

computing the search direction. A few of the search directions proposed by Polak could be viewed as tilted SQP directions (with proper choice of the matrices encapsulating the second-order information in the defining equations). Even with the second-order information, though, it was not possible to guarantee superlinear convergence because no mechanism was included for controlling the *amount* of tilting.

A straightforward way to tilt the SQP direction is, of course, to perturb the right-hand side of the constraints in $QP^0(x, H)$ directly. Building on this observation, Herskovits and Carvalho [25] and Panier and Tits [48] independently developed similar feasible SQP algorithms in which the size of the perturbation was a function of the norm of $d^0(x, H)$ at the current point $x \in X$. Thus, their algorithms required the solution of $QP^0(x, H)$ in order to *define* the perturbed QP. Both algorithms were shown to be superlinearly convergent. On the other hand, as a by-product of the tilting scheme, global convergence proved to be more elusive. In fact, the algorithm in [25] is not globally convergent, while the algorithm in [48] had to resort to a first-order search direction far from a solution in order to guarantee global convergence. Such a hybrid scheme could give slow convergence if a poor initial point is chosen.

The algorithm developed by Panier and Tits in [51], and analyzed under weaker assumptions by Qi and Wei in [64], has enjoyed a great deal of success in practice as implemented in the FFSQP/CFSQP [79, 36] software packages. We will refer to their algorithm as **FSQP**. In [51], instead of directly perturbing $QP^0(x, H)$, tilting is accomplished by replacing d^0 with the convex combination $d = (1-\rho)d^0 + \rho d^1$, where $d^1 \in \mathbb{R}^n$ is an (essentially) arbitrary feasible descent direction (see Figure 2.2). To preserve the local convergence properties of the SQP iteration, $\rho = \rho(d^0) \in [0,1]$ is computed so that d approaches d^0 fast enough (in particular, $\rho(d^0) = O(\|d^0\|^2)$) as the solution is approached. It is suggested that, for example, $d^1 = d^1(x)$ may be taken as a solution of the QP

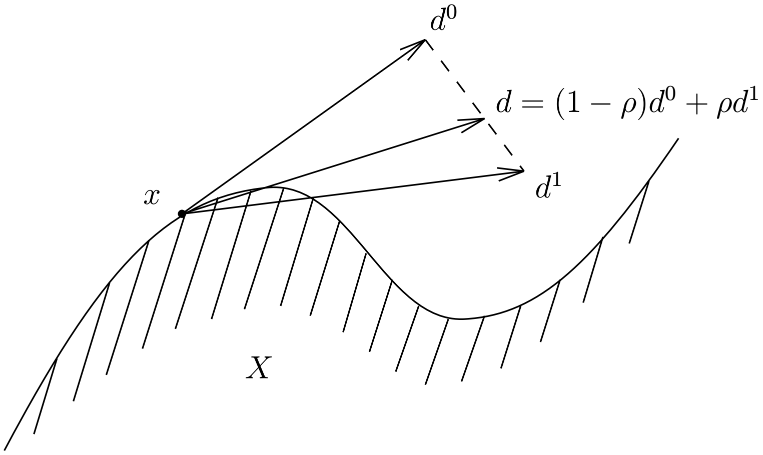

Figure 2.2: “Tilting” the SQP direction d^0 in **FSQP**.

$$\min \ \frac{1}{2}\|d^1\|^2 + \max\left\{\langle \nabla f(x), d^1\rangle + \max_{j=1,\dots,m}\{g_j(x) + \langle \nabla g_j(x), d^1\rangle\}\right\}. \quad QP^1(x)$$

Finally, in order to avoid the Maratos effect and guarantee a superlinear rate of convergence, a second order correction $\tilde{d} = \tilde{d}(x, d, H) \in \mathbb{R}^n$ is used to "bend" the search direction (see Figure 2.3). That is, an Armijo-type search is performed along the arc $x + td + t^2\tilde{d}$. In [51], the Maratos correction $\tilde{d}_k$ is taken as the solution of the QP

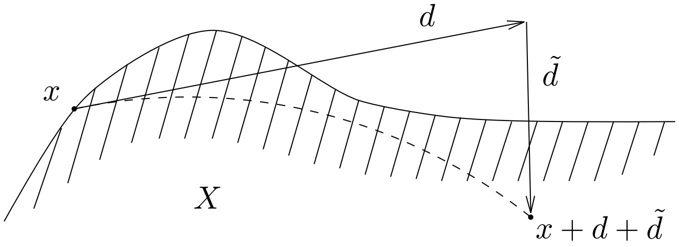

Figure 2.3: "Bending" the direction d in **FSQP**.

$$\begin{array}{ll} \min & \frac{1}{2}\langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d})\rangle + \langle \nabla f(x_k), \hat{d}_k + \tilde{d}\rangle \\ \text{s.t.} & g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \hat{d}_k + \tilde{d}\rangle \leq -\|\hat{d}_k\|^\tau, \quad j = 1, \ldots, m, \end{array} \qquad \widetilde{QP}(x_k, \hat{d}_k, H_k)$$

if it exists and has norm less than $\min\{\|\hat{d}_k\|, C\}$, where $\tau \in (2, 3)$ and C large are given. Otherwise, $\tilde{d}_k = 0$. It is observed in [51] that $\tilde{d}$ could instead be taken as the solution of a linear least squares problem without affecting the asymptotic convergence properties.

From the point of view of computational cost, the main drawback of algorithm **FSQP** is the need to solve three QPs (or two QPs and a linear least squares problem) at each iteration. Clearly, for many problems it would be desirable to reduce the number of QPs at each iteration while preserving the generation of feasible iterates as well as the global and local convergence properties. This is especially critical in the context of those large-scale nonlinear programs for which the time spent solving the QPs dominates that used to evaluate the functions.

Recently there has been a great deal of interest in interior point algorithms for nonconvex nonlinear programming (see, e.g., [11, 13, 74, 8, 52, 73]). Such algorithms generate feasible iterates and typically only require the solution of linear systems of equations in order to generate new iterates. Performance of interior point algorithms tends to be closely related to the careful iterative reduction of a barrier parameter. Essentially, search directions are computed based upon quadratic models of logarithmic barrier functions. On the other hand, SQP-type methods, such as the algorithm proposed here, base search directions upon a quadratic model of the original problem. Thus SQP-type methods should, in general, generate better search directions than interior point methods at the expense of possibly more work per iteration. Of course, work is still very much in its infancy for interior point nonconvex nonlinear programming algorithms. Eventually, such algorithms may be an attractive alternative, especially for very large problems.

Chapter 3

Basic Algorithm

3.1 Introduction

In this chapter we propose and analyze an algorithm to solve the standard smooth nonlinear programming problem (P), which we again repeat for convenience,

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{array} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. The algorithm and analysis of this chapter represent the under-pinnings of this dissertation. In subsequent chapters we will extend the algorithm presented here to handle generalizations of (P).

Recall that the feasible set for (P) is denoted by

$$X \overset{\Delta}{=} \{\ x \in \mathbb{R}^n \mid g_j(x) \leq 0, \quad j = 1, \ldots, m\ \}.$$

We consider a perturbation of $QP^0(x, H)$, the QP used to compute the standard SQP direction (see Chapter 2), which allows us to control the tilting into the feasible set. Specifically, given $x \in X$, $0 < H = H^T \in \mathbb{R}^{n \times n}$, and $0 \leq \eta \in \mathbb{R}$, let

$(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta), \hat{\gamma}(x, H, \eta)) \in \mathbb{R}^n \times \mathbb{R}$ solve the QP

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle \hat{d}, H\hat{d}\rangle + \hat{\gamma} \\ \text{s.t.} \quad & \langle \nabla f(x), \hat{d}\rangle \leq \hat{\gamma}, \\ & g_j(x) + \langle \nabla g_j(x), \hat{d}\rangle \leq \hat{\gamma} \cdot \eta, \quad j = 1, \ldots, m. \end{aligned} \qquad \widehat{QP}(x, H, \eta)$$

In Section 3.3, we show that $\hat{d}$ is a descent direction and, for $\eta > 0$, $\hat{d}$ is a feasible direction. Note that for $\eta \equiv 1$, the search direction is a special case of those computed in Polak's second-order feasible direction algorithms (see Section 4.4 in the book [53]). Further, it is not difficult to show that when $\eta \equiv 0$, we recover the SQP direction, i.e. $\hat{d}(x, H, 0) = d^0(x, H)$. Large values of the parameter η, which we will call the *tilting parameter*, emphasize feasibility, while small values of η emphasize descent.

In [3], Birge, Qi, and Wei propose an SQP algorithm based on $\widehat{QP}(x, H, \eta)$ which generates feasible iterates. Their motivation for introducing the right-hand-side constraint perturbation and the tilting parameters (they use a vector of parameters, one for each constraint) is, like us, to obtain a feasible search direction. Specifically, motivated by the nature of the application problems they are interested in tackling, their goal is to ensure a full step of one is accepted in the line search as early as is possible (so that costly line searches are avoided for most iterations). To this end, their tilting parameters start out positive and, if anything, increase when a step of one is not accepted. A side-effect of such an updating scheme is that the algorithm cannot achieve a superlinear rate of convergence, as the authors point out in Remark 5.1 of [3].

In the present chapter, our goal is to compute a feasible descent direction which approaches the true SQP direction fast enough so as to ensure superlinear convergence. Furthermore, we would like to do this with as little computation per iteration as possible. While computationally the most expensive, algorithm **FSQP** of [51] (see also Section 2.4) has the convergence properties and practical performance we seek. Motivated by this observation, we examine the relevant properties of the search directions generated by algorithm **FSQP** on the sequence of iterates $\{x_k\}$. For $x \in X$, define

$$I(x) \overset{\Delta}{=} \{ \ j \mid g_j(x) = 0 \ \},$$

the index set of active constraints at the point x. In [51], in order for the line-search (with the objective function $f(x)$ used directly as the merit function) to be well-defined, and in order to preserve global and fast local convergence, the sequence of search directions $\{d_k\}$ generated by algorithm **FSQP** is constructed so that the following properties hold:

1. $d_k = 0$ if x_k is a KKT point for (P),
2. $\langle \nabla f(x_k), d_k \rangle < 0$ if x_k is not a KKT point,
3. $\langle \nabla g_j(x_k), d_k \rangle < 0$, for all $j \in I(x_k)$ if x_k is not a KKT point, and
4. $d_k = d_k^0 + O(\|d_k^0\|^2)$.

We will show in Section 3.3 that for $H_k = H_k^T > 0$ and $\eta_k \geq 0$, $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k)$ automatically satisfies the first two properties. Furthermore, $\hat{d}_k$ satisfies the third property if $\eta_k > 0$. Ensuring the fourth property is satisfied requires a bit more care.

In the algorithm presented in Section 3.2, at iteration k, we compute the search direction via $\widehat{QP}(x_k, H_k, \eta_k)$ and the tilting parameter η_k is iteratively adjusted to ensure the four properties are satisfied. The resultant algorithm will be shown to be locally superlinearly convergent and globally convergent without resorting to a first-order direction far from the solution (as is required in the similar scheme proposed in [48]). Further, the generation of a new iterate will only require the solution of one QP and two closely related linear least squares problems. Note that, in contrast with the algorithm presented in [3], our tilting parameter starts out positive and asymptotically approaches zero.

In Section 3.2, we present the details of our new FSQP algorithm. In Section 3.3, we show that under mild assumptions our iteration is globally convergent, as well as locally superlinearly convergent. The algorithm has been implemented and tested and we show in Section 3.4 that the numerical results are quite promising.

3.2 Algorithm

We begin by making a few assumptions that will be in force throughout the chapter.

Assumption 1: The set X is non-empty.

Assumption 2: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable.

Assumption 3: For all $x \in X$ with $I(x) \neq \emptyset$, the set $\{\nabla g_j(x) \mid j \in I(x)\}$ is linearly independent.

Recall that (simplifying (2.1) to the case $p = 1$) a point $x \in \mathbb{R}^n$ is said to be a *Karush-Kuhn-Tucker (KKT)* point for the problem (P) if there exist scalars

(*KKT multipliers*) λ^j, $j = 1, \ldots, m$, satisfying

$$\begin{cases} \nabla f(x) + \displaystyle\sum_{j=1}^{m} \lambda^j \nabla g_j(x) = 0, \\ g_j(x) \leq 0, \quad j = 1, \ldots, m, \\ \lambda^j g_j(x) = 0 \text{ and } \lambda^j \geq 0, \quad j = 1, \ldots, m. \end{cases} \tag{3.1}$$

It is well known that, under our assumptions, a necessary condition for optimality for a point $x \in X$ is that it be a KKT point, i.e. satisfy the KKT conditions.

Note that, with $x \in X$, $\widehat{QP}(x, H, \eta)$ is always consistent: $(0, 0)$ satisfies the constraints. Indeed, $\widehat{QP}(x, H, \eta)$ always has a unique solution $(\hat{d}, \hat{\gamma})$ (see Lemma 1 below) which, by convexity, is its unique KKT point; i.e. there exist multipliers $\hat{\mu} \in \mathbb{R}$ and $\hat{\lambda}^j$, $j = 1, \ldots, m$, which, together with $(\hat{d}, \hat{\gamma})$, satisfy

$$\begin{cases} \begin{bmatrix} H\hat{d} \\ 1 \end{bmatrix} + \hat{\mu} \begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix} + \displaystyle\sum_{j=1}^{m} \hat{\lambda}^j \begin{bmatrix} \nabla g_j(x) \\ -\eta \end{bmatrix} = 0, \\ \langle \nabla f(x), \hat{d} \rangle \leq \hat{\gamma}, \\ g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle \leq \hat{\gamma} \cdot \eta, \quad \forall j = 1, \ldots, m, \\ \hat{\mu} \left(\langle \nabla f(x), \hat{d} \rangle - \hat{\gamma} \right) = 0 \text{ and } \hat{\mu} \geq 0, \\ \hat{\lambda}^j \left(g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle - \hat{\gamma} \cdot \eta \right) = 0 \text{ and } \hat{\lambda}^j \geq 0, \quad \forall j = 1, \ldots, m. \end{cases} \tag{3.2}$$

A simple consequence of the first equation in (3.2), which will be used throughout our analysis, is an affine relationship amongst the multipliers, namely

$$\hat{\mu} + \eta \cdot \sum_{j=1}^{m} \hat{\lambda}^j = 1. \tag{3.3}$$

The parameter η will be iteratively adjusted, i.e. $\eta = \eta_k$, to ensure that $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k)$ has the necessary properties. At iteration k, choosing $\eta_k > 0$ is sufficient to guarantee the first three properties discussed in Section 3.1 are satisfied. As it turns out, though, we will need something a little stronger than this. In order to ensure that, away from a solution, there is adequate tilting into the feasible set (hence the step size does not collapse) we strengthen the positivity requirement to force η_k to be bounded away from zero away from KKT points of (P). Finally, the fourth property requires that $\eta_k \to 0$, as $k \to \infty$, sufficiently fast as $d^0(x_k, H_k) \to 0$. Of course, we do not want to compute $d_k^0 = d^0(x_k, H_k)$, as is done in [48], so we must rely on some other information to update η_k.

Given an estimate I_k of the active set $I(x_k)$, we can compute an estimate $\widehat{d_k^0} = \widehat{d^0}(x_k, H_k, I_k)$ of $d^0(x_k, H_k)$ by solving the *equality* constrained QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle \widehat{d^0}, H_k \widehat{d^0} \rangle + \langle \nabla f(x_k), \widehat{d^0} \rangle \\ \text{s.t.} & g_j(x_k) + \langle \nabla g_j(x_k), \widehat{d^0} \rangle = 0, \quad j \in I_k, \end{array} \qquad LS^0(x_k, H_k, I_k)$$

which is equivalent (after a change of variables) to a linear least squares problem.[1] Let $\hat{I}_k$ be the set of active constraints, not including the "objective descent" constraint $\langle \nabla f(x_k), \hat{d}_k \rangle \le \hat{\gamma}_k$, for $\widehat{QP}(x_k, H_k, \eta_k)$, i.e.

$$\hat{I}_k \triangleq \{\ j \mid g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k \rangle = \hat{\gamma}_k \cdot \eta_k\ \}.$$

We will show in Section 3.3 that $\widehat{d^0}(x_k, H_k, \hat{I}_{k-1}) = d^0(x_k, H_k)$ for all k sufficiently large. Furthermore, it will be shown that, when $\hat{d}_k$ is small, choosing

$$\eta_k \propto \|\widehat{d^0}(x_k, H_k, \hat{I}_{k-1})\|^2$$

will be sufficient to establish global and 2-step superlinear convergence. Proper choice of the proportionality constant (C_k in the algorithm statement below), while not important in the convergence analysis, is critical for satisfactory numerical performance. This will be discussed in Section 3.4.

[1] Which is, in turn, equivalent to a square system of linear equations in $n + |\hat{I}_k^0|$ variables.

In Section 2.4, it was mentioned that a linear least squares problem could be used instead of a QP to compute a version of the Maratos correction $\tilde{d}$ with the same asymptotic convergence properties. Given that our goal is to reduce the computational cost per iteration, it makes sense to use such an approach here. Thus, we take $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{I}_k)$ as the solution, if it exists and is not too large, of the equality constrained QP (equivalent to a least squares problem after a change of variables)

$$\begin{array}{ll} \min & \frac{1}{2}\langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d})\rangle + \langle \nabla f(x_k), \hat{d}_k + \tilde{d}\rangle \\ \text{s.t.} & \quad g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d}\rangle = -\|\hat{d}_k\|^\tau, \qquad \forall j \in \hat{I}_k, \end{array} \qquad \widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{I}_k)$$

where $\tau \in (2,3)$, a direct extension of an alternative considered in [48]. Such an objective, as compared to the pure least squares objective $\|\tilde{d}\|^2$, should improve numerical performance without significantly increasing computational requirements (or affecting the convergence analysis). In the case that $\widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{I}_k)$ is inconsistent, or the computed solution $\tilde{d}_k$ is too large, we could simply set $\tilde{d}_k = 0$. Note that one should use $\widetilde{QP}(x_k, \hat{d}_k, H_k)$ (see Section 2.4) for problems in which function evaluations are expensive compared to the solution of a QP since it provides a better model of (P).

The proposed algorithm is as follows.

Algorithm FSQP′

Parameters: $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\tau \in (2,3)$, $\epsilon_\ell > 0$, $0 < \underline{C} \leq \overline{C}$, $\bar{D} > 0$.

Data: $x_0 \in X$, $0 < H_0 = H_0^T \in \mathbb{R}^{n \times n}$, $0 < \eta_0 \in \mathbb{R}$.

Step 0 - Initialization. **set** $k \leftarrow 0$.

Step 1 - Computation of search arc.

(i). **compute** $(\hat{d}_k, \hat{\gamma}_k) = (\hat{d}(x_k, H_k, \eta_k), \hat{\gamma}(x_k, H_k, \eta_k))$, the active set $\hat{I}_k$, and the associated multipliers $\hat{\mu}_k \in \mathbb{R}$, $\hat{\lambda}_k \in \mathbb{R}^m$.

(ii). **if** $(\hat{d}_k = 0)$ **then stop**.

(iii). **compute** $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{I}_k)$ if it exists and satisfies $\|\tilde{d}_k\| \leq \|\hat{d}_k\|$. Otherwise, **set** $\tilde{d}_k = 0$.

Step 2 - Arc search. **compute** t_k, the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq f(x_k) + \alpha t \langle \nabla f(x_k), \hat{d}_k \rangle,$$

$$g_j(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq 0, \quad j = 1, \dots, m.$$

Step 3 - Updates.

(i). **set** $x_{k+1} \leftarrow x_k + t_k\hat{d}_k + t_k^2\tilde{d}_k$.

(ii). **compute** a new symmetric positive definite estimate H_{k+1} to the Hessian of the Lagrangian.

(iii). **select** $C_{k+1} \in [\underline{C}, \overline{C}]$.

- $*$ **if** $(\|\hat{d}_k\| < \epsilon_\ell)$ **then**
    - · **compute**, if possible,[2] $\widehat{d^0_{k+1}} = \widehat{d^0}(x_{k+1}, H_{k+1}, \hat{I}_k)$, and the associated multipliers $\widehat{\lambda^0_{k+1}} \in \mathbb{R}^{|\hat{I}_k|}$.
    - · **if** $\left(\widehat{d^0_{k+1}}\right.$ exists and $\|\widehat{d^0_{k+1}}\| \leq \bar{D}$ and $\left.\widehat{\lambda^0_{k+1}} \geq 0\right)$ **then set**

    $$\eta_{k+1} \leftarrow C_{k+1} \cdot \|\widehat{d^0_{k+1}}\|^2.$$

    - · **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \|\hat{d}_k\|^2$.

[2] That is, if $LS^0(x_{k+1}, H_{k+1}, \hat{I}_k)$ is non-degenerate.

- $*$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \epsilon_\ell^2$.

(iv). **set** $k \leftarrow k+1$ and **goto** *Step 1.*

3.3 Convergence Analysis

Much of our analysis, especially the local analysis, will be devoted to establishing the relationship between $\hat{d}(x, H, \eta)$ and the SQP direction $d^0(x, H)$. As a consequence, we will be referring to the KKT conditions for $QP^0(x, H)$ in several places. The direction $d^0 = d^0(x, H)$ is a KKT point for $QP^0(x, H)$ if there exists a multiplier $\lambda^0 \in \mathbb{R}^m$ satisfying

$$\begin{cases} Hd^0 + \nabla f(x) + \displaystyle\sum_{j=1}^{m} \lambda^{0,j} \nabla g_j(x) = 0, \\ g_j(x) + \langle \nabla g_j(x), d^0 \rangle \leq 0, \quad j = 1, \dots, m, \\ \lambda^{0,j} \cdot (g_j(x) + \langle \nabla g_j(x), d^0 \rangle) = 0 \text{ and } \lambda^{0,j} \geq 0, \quad j = 1, \dots, m. \end{cases} \tag{3.4}$$

Further, an estimate $\widehat{d^0} = \widehat{d^0}(x, H, I)$ is a KKT point for $LS^0(x, H, I)$ if there exists a multiplier $\widehat{\lambda^0} \in \mathbb{R}^m$ satisfying

$$\begin{cases} H\widehat{d^0} + \nabla f(x) + \displaystyle\sum_{j \in I} \widehat{\lambda^0}^j \nabla g_j(x) = 0, \\ g_j(x) + \langle \nabla g_j(x), \widehat{d^0} \rangle = 0, \quad j \in I. \end{cases} \tag{3.5}$$

Note that the components of $\widehat{\lambda^0}$ for $j \notin I$ play no role in the optimality conditions. We chose to always use $\widehat{\lambda^0} \in \mathbb{R}^m$, independent of the size of I, for notational convenience and consistency in indexing.

3.3.1 Global Convergence

In this section we establish that, under mild assumptions, our proposed algorithm **FSQP′** generates a sequence of iterates $\{x_k\}$ with the property that all accumulation points are KKT points for the problem (P). We begin by establishing some properties of the tilted SQP search direction $\hat{d}(x, H, \eta)$.

Lemma 1. *Given $H = H^T > 0$, $x \in X$, and $\eta \geq 0$, $\hat{d}(x, H, \eta)$ is well-defined and $(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta), \hat{\gamma}(x, H, \eta))$ is the unique KKT point of $\widehat{QP}(x, H, \eta)$. Furthermore, suppose $\{x_k\}_{k \in \mathbb{N}} \subset X$ is bounded, $\{H_k\}_{k \in \mathbb{N}}$ is bounded away from singularity, and $\{\eta_k\}_{k \in \mathbb{N}} \subset [0, \infty)$. Then $\{\hat{d}(x_k, H_k, \eta_k)\}_{k \in \mathbb{N}}$ is bounded.*

Proof. First note that the feasible set for $\widehat{QP}(x, H, \eta)$ is non-empty, since $(\hat{d}, \hat{\gamma}) = (0, 0)$ is always feasible. Now consider the cases $\eta = 0$ and $\eta > 0$ separately. From (3.2) and (3.4), it is clear that, if $\eta = 0$, then $(\hat{d}, \hat{\gamma})$ is a solution to $\widehat{QP}(x, H, 0)$ if, and only if, $\hat{d}$ is a solution of $QP^0(x, H)$ and $\hat{\gamma} = \langle \nabla f(x), \hat{d} \rangle$. It is well known that, under our assumptions, $d^0(x, H)$ is well-defined, unique, and continuous as a function of x. Thus, the lemma follows immediately for this case. Suppose now that $\eta > 0$. In this case, $(\hat{d}, \hat{\gamma})$ is a solution of $\widehat{QP}(x, H, \eta)$ if, and only if, $\hat{d}$ solves the unconstrained problem

$$\min \frac{1}{2}\langle \hat{d}, H\hat{d} \rangle + \max\left\{ \langle \nabla f(x), \hat{d} \rangle,\ \frac{1}{\eta} \cdot \max_{j=1,\dots,m} \{ g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle \} \right\}. \quad (3.6)$$

and

$$\hat{\gamma} = \max\left\{ \langle \nabla f(x), \hat{d} \rangle,\ \frac{1}{\eta} \cdot \max_{j=1,\dots,m} \{ g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle \} \right\}.$$

Since the function being minimized in (3.6) is strictly convex and radially unbounded, it follows that $(\hat{d}(x, H, \eta), \hat{\gamma}(x, H, \eta))$ is well-defined and unique as a global minimizer for the convex problem $\widehat{QP}(x, H, \eta)$, and thus unique as a KKT point for that problem.

To prove the third claim, let $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k)$ and note that since $\{H_k\}_{k\in\mathbb{N}}$ is bounded away from singularity and $H_k = H_k^T > 0$, for all k, there exists $\sigma_1 > 0$ such that

$$\langle \hat{d}_k, H_k \hat{d}_k \rangle \geq \sigma_1 \|\hat{d}_k\|^2, \quad \forall k.$$

Further, the optimal value of the $\widehat{QP}(x_k, H_k, \eta_k)$ is non-positive (since $(0, 0)$ is always feasible), thus

$$\hat{\gamma}_k \leq -\frac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle,$$

for all k. In view of the first QP constraint,

$$\begin{aligned} \langle \nabla f(x_k), \hat{d}_k \rangle &\leq -\frac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle \\ &\leq -\frac{\sigma_1}{2}\|\hat{d}_k\|^2, \end{aligned}$$

for all k. It follows that

$$\|\hat{d}_k\| \leq \frac{2}{\sigma_1}\|\nabla f(x_k)\|,$$

where we have used the inequality $-\|\nabla f(x_k)\|\|\hat{d}_k\| \leq \langle \nabla f(x_k), \hat{d}_k \rangle$. Boundedness of $\{x_k\}_{k\in\mathbb{N}}$ and Assumption 2 gives the result. □

Lemma 2. *Given $H = H^T > 0$ and $\eta \geq 0$*

(i). $\hat{\gamma}(x, H, \eta) \leq 0$ for all $x \in X$. Moreover, $\hat{\gamma}(x, H, \eta) = 0$ if, and only if, $\hat{d}(x, H, \eta) = 0$.

(ii). $\hat{d}(x, H, \eta) = 0$ if, and only if, x is a KKT point for (P).

Proof. To prove (i), note that $(\hat{d}, \hat{\gamma}) = (0, 0)$ is always feasible for $\widehat{QP}(x, H, \eta)$, thus the optimal value of the QP is non-positive. Further, since $H > 0$, the quadratic term in the objective is non-negative, which implies $\hat{\gamma}(x, H, \eta) \leq 0$. Now suppose $\hat{d}(x, H, \eta) = 0$, then feasibility of the first QP constraint implies $\hat{\gamma}(x, H, \eta) = 0$. Finally, suppose $\hat{\gamma}(x, H, \eta) = 0$. Since $x \in X$, $H > 0$, and $\eta \geq 0$, it is clear that $\hat{d} = 0$ is both feasible and achieves the minimum value of the objective. Thus, uniqueness gives $\hat{d}(x, H, \eta) = 0$ and part (i) is proved.

Suppose now that $\hat{d}(x, H, \eta) = 0$. Then $\hat{\gamma}(x, H, \eta) = 0$ and by (3.2) there exist multipliers $\hat{\lambda} \in \mathbb{R}^m$ and $0 \leq \hat{\mu} \in \mathbb{R}$ satisfying

$$\begin{cases} \hat{\mu}\nabla f(x) + \displaystyle\sum_{j=1}^{m} \hat{\lambda}^j \nabla g_j(x) = 0, \\ g_j(x) \leq 0, \quad \forall j = 1, \dots, m, \\ \hat{\lambda}^j g_j(x) = 0 \text{ and } \hat{\lambda}^j \geq 0, \quad \forall j = 1, \dots, m. \end{cases}$$

We begin by showing that $\hat{\mu} > 0$. Proceeding by contradiction, suppose $\hat{\mu} = 0$, then by (3.3) we have

$$\sum_{j=1}^{m} \hat{\lambda}^j > 0.$$

Note that,

$$\begin{aligned} \hat{I} &\triangleq \{\, j \mid g_j(x) + \langle \nabla g_j(x), \hat{d}(x, H, \eta) \rangle = \hat{\gamma}(x, H, \eta) \cdot \eta \,\} \\ &= \{\, j \mid g_j(x) = 0 \,\} = I(x). \end{aligned}$$

Thus, by the complementary slackness condition of (3.2) and the optimality conditions above,

$$0 = \sum_{j=1}^{m} \hat{\lambda}^j \nabla g_j(x) = \sum_{j \in I(x)} \hat{\lambda}^j \nabla g_j(x).$$

By Assumption 3, if $I(x) \neq \emptyset$, then this sum vanishes only if $\hat{\lambda}^j = 0$, for all $j \in I(x)$, but we saw above that this is not the case. Hence we have a contradiction and it follows that $\hat{\mu} > 0$. It is now immediate that x is a KKT point for (P) with multipliers $\lambda^j = \hat{\lambda}^j / \hat{\mu}$, $j = 1, \dots, m$.

Finally, to prove the necessity portion of part (ii) note that if x is a KKT point for (P), then (3.1) shows that $(\hat{d}, \hat{\gamma}) = (0, 0)$ is a KKT point for $\widehat{QP}(x, H, \eta)$, with $\hat{\mu} = (1 + \eta \sum_j \lambda_j)^{-1}$ and $\hat{\lambda}_j = \lambda_j(1 + \eta \sum_j \lambda_j)^{-1}$, $j = 1, \ldots, m$. Uniqueness of such points (Lemma 1) gives the result. □

The next two lemmas establish that the line search in *Step 2* of Algorithm **FSQP′** is well defined.

Lemma 3. *Suppose $x \in X$ is not a KKT point for (P), $H = H^T > 0$, and $\eta > 0$. Then*

(i). $\langle \nabla f(x), \hat{d}(x, H, \eta) \rangle < 0$, and

(ii). $\langle \nabla g_j(x), \hat{d}(x, H, \eta) \rangle < 0$, for all $j \in I(x)$.

Proof. Both follow immediately from Lemma 2 and the fact that $\hat{d}(x, H, \eta)$ and $\hat{\gamma}(x, H, \eta)$ must satisfy the constraints in $\widehat{QP}(x, H, \eta)$. □

Lemma 4. *If $\eta_k = 0$, then x_k is a KKT point for (P) and the algorithm will stop in Step 1(ii) at iteration k. On the other hand, whenever the algorithm does not stop in Step 1(ii), the line search is well defined, i.e. Step 2 yields a step $t_k = \beta^j$ for some finite $j = j(k)$.*

Proof. Suppose that $\eta_k = 0$. Then $k > 0$ and, by *Step 3(iii)*, either $\widehat{d_k^0} = 0$ with $\widehat{\lambda_k^0} \geq 0$, or $\hat{d}_{k-1} = 0$. The latter case cannot hold, as the stopping criterion in *Step 1(ii)* would have stopped the algorithm at iteration $k - 1$. On the other hand, if $\widehat{d_k^0} = 0$ with $\widehat{\lambda_k^0} \geq 0$, then in view of the optimality conditions (3.5), and the fact that x_k is always feasible for (P), we see that x_k is a KKT point for (P) with multipliers

$$\lambda^j = \begin{cases} \widehat{\lambda_k^0}^{\,j}, & j \in \hat{I}_{k-1}, \\ \\ 0, & \text{otherwise.} \end{cases}$$

Thus, by Lemma 2, $\hat{d}_k = 0$ and the algorithm will stop in *Step 1(ii)*. The first claim is thus proved. Also, we have established that $\eta_k > 0$ whenever *Step 2* is reached. The second claim now follows immediately from Lemma 3 and Assumption 2. □

The previous lemmas imply that the algorithm is well-defined. In addition, Lemma 2 shows that if Algorithm **FSQP′** generates a finite sequence terminating at the point x_N, then x_N is a KKT point for the problem (P). We now concentrate on the case in which an infinite sequence $\{x_k\}$ is generated, i.e. the algorithm never satisfies the termination condition in *Step 1(ii)*. Note that, in view of Lemma 4, we may assume throughout that

$$\eta_k > 0, \quad \forall k \in \mathbb{N}. \tag{3.7}$$

Given an infinite index set $\mathcal{K}$, we will use the notation

$$x_k \xrightarrow{k\in\mathcal{K}} x^*$$

to mean

$$x_k \to x^* \ \text{ as } \ k \to \infty, \ k \in \mathcal{K}.$$

Lemma 5. *Suppose $\mathcal{K} \subseteq \mathbb{N}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}} x^* \in X$, $H_k \xrightarrow{k\in\mathcal{K}} H^* > 0$, $\{\eta_k\}$ is bounded on $\mathcal{K}$, and $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Then $\hat{I}_k \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large and the QP multiplier sequences $\{\hat{\mu}_k\}$ and $\{\hat{\lambda}_k\}$ are bounded on $\mathcal{K}$. Further, given any accumulation point $\eta^* \geq 0$ of $\{\eta_k\}_{k\in\mathcal{K}}$, $(0,0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*)$.*

Proof. It follows immediately from non-negativity and (3.3) that $\{\hat{\mu}_k\}_{k\in\mathcal{K}}$ is bounded. Assumption 2 allows us to conclude that $\{\nabla f(x_k)\}_{k\in\mathcal{K}}$ is bounded.

Lemma 2 and the first constraint in $\widehat{QP}(x_k, H_k, \eta_k)$ give

$$\langle \nabla f(x_k), \hat{d}_k \rangle \leq \hat{\gamma}_k \leq 0, \quad \forall k \in \mathcal{K}.$$

Thus, $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} 0$. Next, we will show that $\hat{I}_k \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large. Consider $j' \notin I(x^*)$. There exists $\delta_{j'} > 0$ such that $g_{j'}(x_k) \leq -\delta_{j'} < 0$, for all $k \in \mathcal{K}$, k sufficiently large. In view of Assumption 2, and since $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$, $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} 0$, and $\{\eta_k\}$ is bounded on $\mathcal{K}$, it is clear that

$$g_{j'}(x_k) + \langle \nabla g_{j'}(x_k), \hat{d}_k \rangle - \hat{\gamma}_k \cdot \eta_k \leq -\frac{\delta_{j'}}{2} < 0,$$

i.e. $j' \notin \hat{I}_k$, for all $k \in \mathcal{K}$, k sufficiently large. Hence, $\hat{I}_k \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large, which proves the first claim of the lemma.

Boundedness of $\{\hat{\mu}_k\}_{k\in\mathcal{K}}$ has been proved. To prove that of $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$, using complementary slackness, and the first equation in (3.2), write

$$H_k\hat{d}_k + \hat{\mu}_k \nabla f(x_k) + \sum_{j\in I(x^*)} \hat{\lambda}_k^j \nabla g_j(x_k) = 0. \tag{3.8}$$

Proceeding by contradiction, suppose that $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is unbounded. Without loss of generality, assume that $\|\hat{\lambda}_k\|_\infty > 0$, for all $k \in \mathcal{K}$ and define for all $k \in \mathcal{K}$

$$\nu_k^j \triangleq \frac{\hat{\lambda}_k^j}{\|\hat{\lambda}_k\|_\infty} \in [0, 1].$$

Note that, for all $k \in \mathcal{K}$, $\|\nu_k\|_\infty = 1$. Dividing (3.8) by $\|\hat{\lambda}_k\|_\infty$ and taking limits on an appropriate subsequence of $\mathcal{K}$, it follows that

$$\sum_{j\in I(x^*)} \nu^{*,j} \nabla g_j(x^*) = 0,$$

for some $\nu^{*,j}$, $j \in I(x^*)$, where $\|\nu^*\|_\infty = 1$. As this contradicts Assumption 3, it is established that $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is bounded.

To complete the proof, let $\mathcal{K}' \subseteq \mathcal{K}$ be an infinite index set such that $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^*$ and assume without loss of generality that $\hat{\mu}_k \xrightarrow{k\in\mathcal{K}'} \hat{\mu}^*$ and $\hat{\lambda}_k \xrightarrow{k\in\mathcal{K}'} \hat{\lambda}^*$. Taking limits in the optimality conditions (3.2) shows that, indeed, $(\hat{d}, \hat{\gamma}) = (0, 0)$ is a KKT point for $\widehat{QP}(x^*, H^*, \eta^*)$ with multipliers $\hat{\mu}^*$ and $\hat{\lambda}^*$. Finally, uniqueness of such points (Lemma 1) proves the result. □

Before proceeding, we make an assumption concerning the estimates H_k of the Hessian of the Lagrangian.

Assumption 4: There exist constants $0 < \sigma_1 \leq \sigma_2$ such that, for all k,

$$\sigma_1\|d\|^2 \leq \langle d, H_k d\rangle \leq \sigma_2\|d\|^2, \quad \forall d \in \mathbb{R}^n.$$

Lemma 6. *The sequences $\{H_k\}$ and $\{\eta_k\}$ generated by Algorithm* **FSQP′** *are bounded. Further, the sequence $\{\hat{d}_k\}$ is bounded on subsequences on which $\{x_k\}$ is bounded.*

Proof. That $\{H_k\}$ is bounded follows immediately from Assumption 4. *Step 3(iii)* of Algorithm **FSQP′** ensures that the sequence $\{\eta_k\}$ is bounded. Finally, it then follows from Lemma 1 that $\{\hat{d}_k\}$ is bounded on subsequences on which $\{x_k\}$ is bounded. □

Lemma 7. *If $\mathcal{K} \subseteq \mathbb{N}$ is an infinite index set such that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$, then all accumulation points of $\{x_k\}_{k\in\mathcal{K}}$ are KKT points for (P).*

Proof. Suppose $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite index set on which $x_k \xrightarrow{k\in\mathcal{K}'} x^* \in X$. In view of Lemma 6, assume, without loss of generality that $H_k \xrightarrow{k\in\mathcal{K}'} H^* > 0$ and $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^* \geq 0$. Lemma 5 shows that $(0, 0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*)$. Thus, in view of Lemma 2, x^* is a KKT point for (P). □

We now state and prove the main result of this section.

Theorem 3. *Under the stated assumptions, Algorithm* **FSQP′** *generates a sequence* $\{x_k\}$ *for which all accumulation points are KKT points for* (P).

Proof. Suppose $\mathcal{K} \subseteq \mathbb{N}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}} x^*$. In view of Lemma 6, we may assume without loss of generality that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} \hat{d}^*$, $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \geq 0$, and $H_k \xrightarrow{k\in\mathcal{K}} H^* > 0$. The cases $\eta^* = 0$ and $\eta^* > 0$ are considered separately.

Suppose first that $\eta^* = 0$. Then, by *Step 3(iii)*, either $\widehat{d_k^0} \xrightarrow{k\in\mathcal{K}} 0$ with $\widehat{\lambda_k^0} \geq 0$, for all $k \in \mathcal{K}$, k large enough, or $\hat{d}_{k-1} \xrightarrow{k\in\mathcal{K}} 0$. If the latter case holds, it is then clear that $x_{k-1} \xrightarrow{k\in\mathcal{K}} x^*$, since $\|x_k - x_{k-1}\| \leq 2\|\hat{d}_{k-1}\| \xrightarrow{k\in\mathcal{K}} 0$. Thus, by Lemma 7, x^* is a KKT point for (P). Now suppose instead that $\widehat{d_k^0} \xrightarrow{k\in\mathcal{K}} 0$ with $\widehat{\lambda_k^0} \geq 0$, for all $k \in \mathcal{K}$, k large enough. Using an argument very similar to that used in Lemma 5, we can show that $\{\widehat{\lambda_k^0}\}_{k\in\mathcal{K}}$ is a bounded sequence and $\hat{I}_{k-1} \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large. Thus, taking limits in (3.5) on an appropriate subsequence of $\mathcal{K}$ shows that x^* is a KKT point for (P).

Now consider the case $\eta^* > 0$. We will show that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Proceeding by contradiction, without loss of generality suppose there exists $\underline{d} > 0$ such that $\|\hat{d}_k\| \geq \underline{d}$ for all $k \in \mathcal{K}$. Thus, from non-positivity of the optimal value of the objective function in $\widehat{QP}(x_k, H_k, \eta_k)$ (since $(0, 0)$ is always feasible) and Assumption 4, we see that

$$\hat{\gamma}_k \leq -\frac{1}{2}\sigma_1 \underline{d}^2 < 0, \quad \forall k \in \mathcal{K}.$$

Further, in view of (3.7) and since $\eta^* > 0$, there exists $\underline{\eta} > 0$ such that

$$\eta_k > \underline{\eta}, \quad \forall k \in \mathcal{K}.$$

From the constraints of $\widehat{QP}(x_k, H_k, \eta_k)$, it follows that

$$\langle \nabla f(x_k), \hat{d}_k \rangle \leq -\frac{1}{2}\sigma_1 \underline{d}^2 < 0, \quad \forall k \in \mathcal{K},$$

and

$$g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k \rangle \leq -\frac{1}{2}\sigma_1 \underline{d}^2 \underline{\eta} < 0, \quad \forall k \in \mathcal{K},$$

$j = 1, \ldots, m$. Hence, using Assumption 2, it is easily shown that there exists $\delta > 0$ such that for all $k \in \mathcal{K}$, k large enough,

$$\langle \nabla f(x_k), \hat{d}_k \rangle \leq -\delta,$$

$$\langle \nabla g_j(x_k), \hat{d}_k \rangle \leq -\delta, \quad \forall j \in I(x^*)$$

$$g_j(x_k) \leq -\delta, \quad \forall j \in \{1, \ldots, m\} \setminus I(x^*).$$

The rest of the contradiction argument establishing $\hat{d}_k \xrightarrow{k \in \mathcal{K}} 0$ follows exactly the proof of Proposition 3.2 in [48]. Finally, it then follows from Lemma 7 that x^* is a KKT point for (P). □

3.3.2 Local Convergence

While the details are often quite different, overall the analysis in this section is inspired by and occasionally follows that of Panier and Tits in [48, 51]. In order to establish a rate of convergence for the algorithm, we first strengthen the regularity assumptions.

Assumption 2′: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are three times continuously differentiable.

Recall that a point x^* is said to satisfy the *second order sufficiency conditions with strict complementary slackness* for (P) if there exists a multiplier vector $\lambda^* \in \mathbb{R}^m$ such that

- The pair (x^*, λ^*) satisfies (3.1), i.e. x^* is a KKT point for (P),

- $\nabla^2_{xx} L(x^*, \lambda^*)$ is positive definite on the subspace

$$\{h \mid \langle \nabla g_j(x^*), h\rangle = 0,\ \forall j \in I(x^*)\},$$

- and $\lambda^{*,j} > 0$ for all $j \in I(x^*)$ (strict complementary slackness).

In order to guarantee that the entire sequence $\{x_k\}$ converges to a KKT point x^*, we make the following assumption. Recall that we have already established, under weaker assumptions, that *every* accumulation point of $\{x_k\}$ is a KKT point for (P).

Assumption 5: The sequence $\{x_k\}$ has an accumulation point x^* which satisfies the second order sufficiency conditions with strict complementary slackness.

It is well known, and not difficult to show, that Assumption 5 guarantees the entire sequence converges. For a proof see, e.g., Proposition 4.1 in [48]. We state the result here without proof.

Lemma 8. *The entire sequence generated by Algorithm* **FSQP$'$** *converges to a point x^* satisfying the second order sufficiency conditions with strict complementary slackness.*

From this point forward, λ^* will denote the (unique) multiplier vector satisfying the KKT conditions for (P) at x^*. Further, we need to strengthen the assumptions concerning the sequence $\{H_k\}$.

Assumption 6: The sequence $\{H_k\}$ converges to some $H^* = H^{*T} > 0$.

In order to establish a rate of convergence, we will show that our sequence of tilted SQP directions approaches the true SQP direction, for which asymptotic rates of convergence are well known, sufficiently fast. In order to do so, define $d_k^0 = d^0(x_k, H_k)$, where x_k and H_k are as computed by Algorithm **FSQP′** . Further, for each k, define $\lambda_k^0 \in \mathbb{R}^m$ as a multiplier vector satisfying (3.4) at d_k^0 and let $I_k^0 \triangleq \{ \ j \mid g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0 \ \}$. The following lemma is proved in [48, 51] under identical assumptions.

Lemma 9.

(i) $d_k^0 \to 0$,

(ii) $\lambda_k^0 \to \lambda^*$.

(iii) For all k *sufficiently large, the following two equalities hold*

$$I_k^0 = \{ \ j \mid \lambda_k^{0,j} > 0 \ \} = I(x^*).$$

Before proceeding, we state one more well-known result that will be called upon several times throughout the balance of the analysis. First, we make the definitions

$$R_k \triangleq [\ \nabla g_j(x_k) \ : \ j \in I(x^*) \ ] \, ,$$
$$g_k \triangleq [\ g_j(x_k) \ : \ j \in I(x^*) \ ]^T .$$

Lemma 10. *Under the stated assumptions, the matrix*

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix}$$

is uniformly invertible, i.e. it has bounded condition number for all k.

We now establish that the entire tilted SQP direction sequence converges to 0. In order to do so, we establish that $\hat{d}(x, H, \eta)$ is continuous in a neighborhood of (x^*, H^*, η^*), for any $\eta^* \geq 0$. Complicating the analysis is the fact that we have yet to establish that the sequence $\{\eta_k\}$ does, in fact, converge. Given $\eta^* \geq 0$, define the set

$$N^*(\eta^*) \triangleq \left\{ \begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix}, \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix}, j \in I(x^*) \right\}.$$

Lemma 11. *Given any $\eta^* \geq 0$, the set $N^*(\eta^*)$ is linearly independent.*

Proof. Note that, in view of Lemma 2, $\hat{d}^* = \hat{d}(x^*, H^*, \eta^*) = 0$. Now suppose the lemma does not hold, i.e. suppose there exist scalars λ^j, $j \in \{0\} \cup I(x^*)$, not all zero, such that

$$\lambda^0 \begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix} + \sum_{j \in I(x^*)} \lambda^j \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix} = 0. \tag{3.9}$$

In view of Assumption 3, $\lambda^0 \neq 0$ and the scalars λ^j are unique modulo a scaling factor. This uniqueness, the fact that $\hat{d}^* = 0$, and the optimality conditions (3.2) imply that $\hat{\mu}^* = 1$ and

$$\hat{\lambda}^{*,j} = \begin{cases} \dfrac{\lambda^j}{\lambda^0} & j \in I(x^*) \\ 0 & \text{else}, \end{cases}$$

$j = 1, \ldots, m$ are KKT multipliers for $\widehat{QP}(x^*, H^*, \eta^*)$. Thus, in view of (3.3),

$$\eta^* \cdot \sum_{j \in I(x^*)} \frac{\lambda^j}{\lambda^0} = 0.$$

But this contradicts (3.9), which gives

$$\eta^* \cdot \sum_{j \in I(x^*)} \frac{\lambda^j}{\lambda^0} = -1,$$

hence $N^*(\eta^*)$ is linearly independent. □

Lemma 12. *Let $\eta^* \geq 0$ be an accumulation point of $\{\eta_k\}$. Then $(\hat{d}^*, \hat{\gamma}^*) = (0,0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*)$ and the second order sufficiency conditions hold, with strict complementary slackness.*

Proof. In view of Lemma 2, $\widehat{QP}(x^*, H^*, \eta^*)$ has $(\hat{d}^*, \hat{\gamma}^*) = (0,0)$ as its unique solution. Define the Lagrangian function $\hat{L}^* : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ for $\widehat{QP}(x^*, H^*, \eta^*)$ as

$$\begin{aligned}\hat{L}^*(\hat{d}, \hat{\gamma}, \hat{\mu}, \hat{\lambda}) \quad = \quad & \frac{1}{2}\langle \hat{d}, H^*\hat{d}\rangle + \hat{\gamma} + \hat{\mu}\left(\langle \nabla f(x^*), \hat{d}\rangle - \hat{\gamma}\right) \\ & + \sum_{j=1}^{m} \hat{\lambda}^j \left(g_j(x^*) + \langle \nabla g_j(x^*), \hat{d}\rangle - \hat{\gamma}\eta^*\right).\end{aligned}$$

Suppose $\hat{\mu}^* \in \mathbb{R}$ and $\hat{\lambda}^* \in \mathbb{R}^m$ are multipliers satisfying (3.2) at $(\hat{d}^*, \hat{\gamma}^*)$. Let $j = 0$ be the index for the first constraint in $\widehat{QP}(x^*, H^*, \eta^*)$, i.e. $\langle \nabla f(x^*), \hat{d}\rangle \leq \hat{\gamma}$. Note that since $(\hat{d}^*, \hat{\gamma}^*) = (0,0)$, the active constraint index set[3] $\hat{I}^*$ for $\widehat{QP}(x^*, H^*, \eta^*)$ is equal to $I(x^*)$, the active constraint index set for (P) at x^*, in addition to $j = 0$. Thus the set of active constraint gradients for $\widehat{QP}(x^*, H^*, \eta^*)$ is $N^*(\eta^*)$.

Now consider the Hessian of the Lagrangian for $\widehat{QP}(x^*, H^*, \eta^*)$, i.e. the second derivative with respect to the first two variables $(\hat{d}, \hat{\gamma})$,

$$\nabla^2 \hat{L}^*(0, 0, \hat{\lambda}^*, \hat{\mu}^*) = \begin{bmatrix} H^* & 0 \\ 0 & 0 \end{bmatrix},$$

and given an arbitrary $h \in \mathbb{R}^{n+1}$, decompose it as $h = (y^T, \alpha)^T$, where $y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. Then clearly,

$$h^T \nabla^2 \hat{L}^*(0, 0, \hat{\lambda}^*, \hat{\mu}^*) h \geq 0, \quad \forall h$$

[3] We are temporarily abandoning our convention of omitting the objective descent constraint in $\hat{I}$ for this argument only.

and for $h \neq 0$, $h^T \nabla^2 \hat{L}^*(0,0,\hat{\lambda}^*,\hat{\mu}^*)h = y^T H^* y$ is zero if, and only if, $y = 0$ and $\alpha \neq 0$. Since

$$\begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix}^T \begin{pmatrix} 0 \\ \alpha \end{pmatrix} = -\alpha \neq 0,$$

it then follows that $\nabla^2 \hat{L}^*(0,0,\hat{\lambda}^*,\hat{\mu}^*)$ is positive definite on $N^*(\eta^*)^\perp$, the tangent space to the active constraints for $\widehat{QP}(x^*, H^*, \eta^*)$ at $(0,0)$. Thus, it is established that the second order sufficiency conditions hold. We next show that strict complementary slackness holds.

First, $\hat{\mu}^* > 0$. Indeed, suppose to the contrary that $\hat{\mu}^* = 0$. In view of (3.3), this implies there exists an index $j' \in \hat{I}^*$ such that $\hat{\lambda}^{*,j'} > 0$. Recalling that $\hat{I}^* = I(x^*) \cup \{0\}$ and invoking complementary slackness for $\widehat{QP}(x^*, H^*, \eta^*)$, the first equation in (3.2) gives

$$\sum_{j \in I(x^*)} \hat{\lambda}^{*,j} \nabla g_j(x^*) = 0.$$

As $\hat{\lambda}^{*,j'} > 0$ for some $j' \in \hat{I}^*$, this contradicts Assumption 3. Next, a well-known consequence of Assumption 3 is that the KKT multipliers $\lambda^{*,j}$ for (P) at x^* are unique. Thus, it follows from the optimality conditions (3.2) and (3.1) that $\hat{\lambda}^{*,j} = \hat{\mu}^* \cdot \lambda^{*,j}$, $j = 1, \ldots, m$. Further, it follows from Assumption 5 that $\hat{\lambda}^{*,j} > 0$, $j \in I(x^*)$, i.e. strict complementary slackness is satisfied by $\widehat{QP}(x^*, H^*, \eta^*)$ at $(0,0)$. □

Lemma 13. *If $\mathcal{K}$ is a subsequence on which $\{\eta_k\}$ converges, say to $\eta^* \geq 0$, then $\hat{\mu}_k \xrightarrow{k \in \mathcal{K}} \hat{\mu}^* > 0$ and $\hat{\lambda}_k \xrightarrow{k \in \mathcal{K}} \hat{\mu}^* \cdot \lambda^*$, where $\hat{\mu}^* = \hat{\mu}^*(\eta^*)$ is the KKT multiplier for the first constraint of $\widehat{QP}(x^*, H^*, \eta^*)$. Finally, $\hat{d}_k \to 0$ and $\hat{\gamma}_k \to 0$.*

Proof. In view of Lemmas 11 and 12, we may invoke a result due to Robinson

(Theorem 2.1 in [68]) to conclude

$$(\hat{d}_k, \hat{\gamma}_k) \xrightarrow{k\in\mathcal{K}} (0,0), \quad \hat{\mu}_k \xrightarrow{k\in\mathcal{K}} \hat{\mu}^*, \quad \text{and} \quad \frac{\hat{\lambda}_k}{\hat{\mu}^*} \xrightarrow{k\in\mathcal{K}} \lambda^*.$$

It is important to note that $\hat{\mu}^*$ is a function of η^*, i.e. $\hat{\mu}^* = \hat{\mu}^*(\eta^*)$. Now suppose that the last claim of the lemma does not hold. If $\hat{d}_k \not\to 0$, there exists an infinite index set $\mathcal{K} \subseteq \mathbb{N}$ and $\underline{d} > 0$ such that $\|\hat{d}_k\| \geq \underline{d}$, for all $k \in \mathcal{K}$. As $\{\eta_k\}_{k\in\mathcal{K}}$ is bounded, there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ and $\eta^* \geq 0$ such that $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^*$. By what we showed above, $\hat{d}_k \xrightarrow{k\in\mathcal{K}'} 0$, which is a contradiction, hence $\hat{d}_k \to 0$. It immediately follows from the first constraint of $\widehat{QP}(x_k, H_k, \eta_k)$ that $\hat{\gamma}_k \to 0$. □

Lemma 14. *For all k sufficiently large, $\hat{I}_k = I(x^*)$.*

Proof. Since $\{\eta_k\}$ is bounded and $(\hat{d}_k, \hat{\gamma}_k) \to (0,0)$, in view of Lemma 5, $\hat{I}_k \subseteq I(x^*)$, for all k sufficiently large. Now suppose it does not hold that $\hat{I}_k = I(x^*)$ for all k sufficiently large. Thus, there exists $j' \in I(x^*)$ and an infinite index set $\mathcal{K} \subseteq \mathbb{N}$ such that $j' \notin \hat{I}_k$, for all $k \in \mathcal{K}$. Now, in view of Lemma 6, there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ and $\eta^* \geq 0$ such that $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^*$. Since $j' \in I(x^*)$, Assumption 5 guarantees $\lambda^{*,j'} > 0$. Further, Lemma 13 shows that $\hat{\lambda}_k^{j'} \xrightarrow{k\in\mathcal{K}'} \hat{\mu}^*(\eta^*) \cdot \lambda^{*,j'} > 0$. Therefore, $\hat{\lambda}_k^{j'} > 0$ for all k sufficiently large, $k \in \mathcal{K}'$, which, by complementary slackness, implies $j' \in \hat{I}_k$ for all $k \in \mathcal{K}'$ large enough. Since $\mathcal{K}' \subseteq \mathcal{K}$, this is a contradiction, hence $\hat{I}_k = I(x^*)$, for all k sufficiently large. □

Given a vector $\lambda \in \mathbb{R}^m$, define the notation

$$\lambda^+ \triangleq [\ \lambda^j \ : \ j \in I(x^*)\ ]^T.$$

Note that, in view of Lemma 9(iii), for k large enough, the optimality conditions (3.4), yield

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^0 \\ (\lambda_k^0)^+ \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix}. \tag{3.10}$$

Lemma 15. *For all k sufficiently large, $\widehat{d_k^0} = d_k^0$.*

Proof. In view of Lemma 14 and the optimality conditions (3.5), the estimate $\widehat{d_k^0}$ and its corresponding multiplier vector $\widehat{\lambda_k^0}$ (recall that for ease of notation we defined $\widehat{\lambda_k^0} \in \mathbb{R}^m$) satisfy

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} \widehat{d_k^0} \\ (\widehat{\lambda_k^0})^+ \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix}, \tag{3.11}$$

for all k sufficiently large. In view of (3.10), the result then follows from Lemma 10. □

Lemma 16.

(i) $\eta_k \to 0$,

(ii) $\hat{\mu}_k \to 1$, *and* $\hat{\lambda}_k \to \lambda^*$.

(iii) For all k sufficiently large, $\hat{I}_k = \{ \; j \mid \hat{\lambda}_k^j > 0 \; \}$.

Proof. Claim (i) follows from *Step 3(iii)* of Algorithm **FSQP′**, since in view of Lemma 13, Lemma 15, and Lemma 9, $\{\hat{d}_k\}$ and $\{\widehat{d_k^0}\}$ both converge to 0. In view of (i), Lemma 13 establishes that $\hat{\mu}_k \to \hat{\mu}^*(0)$, and $\hat{\lambda}_k \to \hat{\mu}^*(0) \cdot \lambda^*$. That $\hat{\mu}^*(0) = 1$ follows from (3.3), hence claim (ii) is proved. Finally, claim (iii) follows from claim (ii), Lemma 14, and Assumption 5. □

We now focus our attention on establishing relationships between $\hat{d}_k$, $\tilde{d}_k$, and the true SQP direction d_k^0.

Lemma 17.

(i) $\eta_k = O(\|d_k^0\|^2)$,

(ii) $\hat{d}_k = d_k^0 + O(\|d_k^0\|^2)$.

(iii) $\hat{\gamma}_k = O(\|d_k^0\|)$.

Proof. In view of Lemma 15, $\widehat{d_k^0}$ exists and $\widehat{d_k^0} = d_k^0$ for all k sufficiently large. Lemmas 13 and 9 ensure that *Step 3(iii)* of Algorithm **FSQP′** chooses $\eta_k = C_k \cdot \|\widehat{d_k^0}\|^2$ for all k sufficiently large, thus (i) follows. It is clear from Lemma 14 and the optimality conditions (3.2) that $\hat{d}_k$ and $\hat{\lambda}_k$ satisfy

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} \hat{d}_k \\ \hat{\lambda}_k^+ \end{pmatrix} = - \begin{pmatrix} \hat{\mu}_k \cdot \nabla f(x_k) \\ g_k - \eta_k \cdot \hat{\gamma}_k \cdot \mathbf{1}_{|I(x^*)|} \end{pmatrix}$$

$$= - \begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix} + \eta_k \cdot \begin{pmatrix} \left(\sum_{j \in I(x^*)} \hat{\lambda}_k^j \right) \cdot \nabla f(x_k) \\ \hat{\gamma}_k \cdot \mathbf{1}_{|I(x^*)|} \end{pmatrix}, \qquad (3.12)$$

for all k sufficiently large, where $\mathbf{1}_{|I(x^*)|}$ is a vector of $|I(x^*)|$ ones. It thus follows from (3.10) that

$$\hat{d}_k = d_k^0 + O(\eta_k),$$

and in view of claim (i), claim (ii) follows. Finally, since (from the QP constraint and Lemma 2) $\langle \nabla f(x_k), \hat{d}_k \rangle \leq \hat{\gamma}_k < 0$, it is clear that $\hat{\gamma}_k = O(\|\hat{d}_k\|) = O(\|d_k^0\|)$. □

Lemma 18. $\tilde{d}_k = O(\|d_k^0\|^2)$.

Proof. Let

$$c_k \triangleq [-g_j(x_k + \hat{d}_k) - \|\hat{d}_k\|^\tau \ : \ j \in I(x^*)]^T.$$

Expanding $g_j(\cdot)$, $j \in I(x^*)$, about x_k we see that, for some $\xi^j \in (0,1)$, $j \in I(x^*)$,

$$c_k = [\ \overbrace{-g_j(x_k) - \langle \nabla g_j(x_k), \hat{d}_k\rangle}^{=-\eta_k \cdot \hat{\gamma}_k} \\ + \frac{1}{2}\langle \hat{d}_k, \nabla^2 g_j(x_k + \xi^j \hat{d}_k)\hat{d}_k\rangle - \|\hat{d}_k\|^\tau \ : \ j \in I(x^*) \ ]^T.$$

Since $\tau > 2$, from Lemma 17, we conclude $c_k = O(\|d_k^0\|^2)$. Now, for all k sufficiently large, $\hat{I}_k = I(x^*)$, $\tilde{d}_k$ is well-defined and satisfies

$$g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d}_k\rangle = -\|\hat{d}_k\|^\tau, \quad j \in I(x^*), \tag{3.13}$$

thus, we have established

$$R_k^T \tilde{d}_k = O(\|d_k^0\|^2). \tag{3.14}$$

The first order KKT conditions for $\widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{I}_k)$ tell us there exists a multiplier $\tilde{\lambda}_k \in \mathbb{R}^{|I(x^*)|}$ satisfying

$$\begin{cases} H_k(\hat{d}_k + \tilde{d}_k) + \nabla f(x_k) + R_k \tilde{\lambda}_k = 0, \\ R_k^T \tilde{d}_k = c_k. \end{cases}$$

Also, from the optimality conditions (3.12) we have

$$H_k \hat{d}_k + \nabla f(x_k) = q_k - R_k \hat{\lambda}_k^+,$$

where

$$q_k \triangleq \eta_k \cdot \left(\sum_{j \in I(x^*)} \hat{\lambda}_k^j \right) \cdot \nabla f(x_k).$$

So, $\tilde{d}_k$ and $\tilde{\lambda}_k$ satisfy

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} \tilde{d}_k \\ \tilde{\lambda}_k \end{pmatrix} = \begin{pmatrix} R_k \hat{\lambda}_k^+ - q_k \\ c_k \end{pmatrix}.$$

Solving for $\tilde{d}_k$, after a little algebra we obtain

$$\begin{aligned}\tilde{d}_k &= H_k^{-1}R_k(R_k^T H_k^{-1}R_k)^{-1}c_k \\ &\quad + \left[H_k^{-1} - H_k^{-1}R_k(R_k^T H_k^{-1}R_k)^{-1}R_k^T H_k^{-1}\right](R_k\hat{\lambda}_k^+ - q_k) \\ &= H_k^{-1}R_k(R_k^T H_k^{-1}R_k)^{-1}c_k \\ &\quad + \left[H_k^{-1} - H_k^{-1}R_k(R_k^T H_k^{-1}R_k)^{-1}R_k^T H_k^{-1}\right](-q_k).\end{aligned}$$

Further, in view of Lemma 17 and boundedness of all sequences, $q_k = O(\|d_k^0\|^2)$. Thus, $\tilde{d}_k$ equivalently satisfies

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix}\begin{pmatrix} \tilde{d}_k \\ \lambda_k' \end{pmatrix} = \begin{pmatrix} -q_k \\ c_k \end{pmatrix} = O(\|d_k^0\|^2),$$

for some $\lambda_k' \in \mathbb{R}^{|I(x^*)|}$. The result then follows from Lemma 10. □

We now add one additional assumption to ensure that the matrices $\{H_k\}$ suitably approximate the Hessian of the Lagrangian at the solution. Define the projection

$$P_k \triangleq I - R_k(R_k^T R_k)^{-1}R_k^T.$$

Assumption 7:

$$\lim_{k\to\infty}\frac{\|P_k(H_k - \nabla^2_{xx}L(x^*,\lambda^*))P_k\hat{d}_k\|}{\|\hat{d}_k\|} = 0.$$

The following technical lemma will be needed in order to establish that eventually the step of one is always accepted by the line search.

Lemma 19. *There exist constants* $\nu_1,\ \nu_2,\ \nu_3 > 0$ *such that*

(i) $\langle\nabla f(x_k), \hat{d}_k\rangle \le -\nu_1\|d_k^0\|^2$,

(ii) for all k sufficiently large

$$\sum_{j=1}^{m} \hat{\lambda}_k^j g_j(x_k) \leq -\nu_2 \|g_k\|,$$

(iii) $\hat{d}_k = P_k \hat{d}_k + d_k^1$, where

$$\|d_k^1\| \leq \nu_3 \|g_k\| + O(\|d_k^0\|^3),$$

for all k sufficiently large.

Proof. To show part (i), note that in view of the first QP constraint, negativity of the optimal value of the QP objective, and Assumption 4,

$$\begin{array}{rcl} \langle \nabla f(x_k), \hat{d}_k \rangle & \leq & \hat{\gamma}_k \\ & \leq & -\frac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle \\ & \leq & -\frac{\sigma_1}{2}\|\hat{d}_k\|^2 = -\frac{\sigma_1}{2}\|d_k^0\|^2 + O(\|d_k^0\|^4). \end{array}$$

The proof of part (ii) is identical to that of Lemma 4.4 in [48]. To show (iii), note that from (3.12) for all k sufficiently large, $\hat{d}_k$ satisfies

$$R_k^T \hat{d}_k = -g_k - \hat{\gamma}_k \eta_k \cdot \mathbf{1}_{|I(x^*)|}.$$

Thus, we can write $\hat{d}_k = P_k \hat{d}_k + d_k^1$, where

$$d_k^1 = -R_k (R_k^T R_k)^{-1} (g_k + \hat{\gamma}_k \eta_k \cdot \mathbf{1}_{|I(x^*)|}).$$

The result follows from Assumption 3. □

Lemma 20. *For all k sufficiently large, $t_k = 1$.*

Proof. Following [48], consider an expansion of $g_j(\cdot)$ about $x_k + \hat{d}_k$ for $j \in I(x^*)$, for all k sufficiently large,

$$\begin{aligned} g_j(x_k + \hat{d}_k + \tilde{d}_k) &= g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k + \hat{d}_k), \tilde{d}_k \rangle + O(\|d_k^0\|^4) \\ &= g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d}_k \rangle + O(\|d_k^0\|^3) \\ &= -\|\hat{d}_k\|^\tau + O(\|d_k^0\|^3) \\ &= -\|d_k^0\|^\tau + O(\|d_k^0\|^3), \end{aligned}$$

where we have used Lemmas 17 and 18, boundedness of all sequences, and the constraints from $\widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{I}_k)$ ($\hat{I}_k = I(x^*)$ for all k sufficiently large by Lemma 14). As $\tau < 3$, it follows that $g_j(x_k + \hat{d}_k + \tilde{d}_k) \leq 0$, $j \in I(x^*)$, for all k sufficiently large. The same result trivially holds for $j \notin I(x^*)$. Further, we have

$$g_j(x_k + \hat{d}_k + \tilde{d}_k) = O(\|d_k^0\|^\tau), \quad j \in I(x^*). \tag{3.15}$$

In view of Assumption 2′ and Lemmas 17 and 18,

$$\begin{aligned} f(x_k + \hat{d}_k + \tilde{d}_k) &= f(x_k) + \langle \nabla f(x_k), \hat{d}_k \rangle + \langle \nabla f(x_k), \tilde{d}_k \rangle \\ &\qquad + \tfrac{1}{2}\langle \hat{d}_k, \nabla^2 f(x_k) \hat{d}_k \rangle + O(\|d_k^0\|^3). \end{aligned}$$

From the optimality conditions (3.2), Lemma 17(i), and boundedness of all sequences, we see

$$H_k \hat{d}_k + \nabla f(x_k) + \sum_{j=1}^{m} \hat{\lambda}_k^j \nabla g_j(x_k) = O(\|d_k^0\|^2). \tag{3.16}$$

Complementary slackness for $\widehat{QP}(x_k, H_k, \eta_k)$ and Lemma 17 yield

$$\hat{\lambda}_k^j \langle \nabla g_j(x_k), \hat{d}_k \rangle = -\hat{\lambda}_k^j g_j(x_k) + O(\|d_k^0\|^3). \tag{3.17}$$

Taking the inner product of (3.16) with $\hat{d}_k$, then adding and subtracting the quantity $\sum_j \hat{\lambda}_k^j \langle \nabla g_j(x_k), \hat{d}_k \rangle$, using (3.17), and finally multiplying the result by

$\frac{1}{2}$ gives

$$\begin{aligned}\tfrac{1}{2}\langle \nabla f(x_k), \hat{d}_k\rangle &= -\frac{1}{2}\langle \hat{d}_k, H_k\hat{d}_k\rangle - \sum_{j=1}^{m} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \hat{d}_k\rangle \\ &\qquad - \frac{1}{2}\sum_{j=1}^{m} \hat{\lambda}_k^j g_j(x_k) + O(\|d_k^0\|^3).\end{aligned}$$

Further, Lemmas 17 and 18 and (3.16) give

$$\langle \nabla f(x_k), \tilde{d}_k\rangle = -\sum_{j=1}^{m} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \tilde{d}_k\rangle + O(\|d_k^0\|^3).$$

Combining results, we have

$$\begin{aligned} & f(x_k + \hat{d}_k + \tilde{d}_k) - f(x_k) = \\ &\qquad \frac{1}{2}\langle \nabla f(x_k), \hat{d}_k\rangle - \frac{1}{2}\langle \hat{d}_k, H_k\hat{d}_k\rangle - \frac{1}{2}\sum_{j=1}^{m} \hat{\lambda}_k^j g_j(x_k) \\ &\qquad - \sum_{j=1}^{m} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \hat{d}_k\rangle - \sum_{j=1}^{m} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \tilde{d}_k\rangle \\ &\qquad + \frac{1}{2}\langle \hat{d}_k, \nabla^2 f(x_k)\hat{d}_k\rangle + O(\|d_k^0\|^3). \end{aligned} \tag{3.18}$$

Expanding about x_k and using Lemmas 17(ii) and 18 and equation (3.15) we have

$$\begin{aligned} & g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k\rangle + \langle \nabla g_j(x_k), \tilde{d}_k\rangle = \\ &\qquad -\frac{1}{2}\langle \hat{d}_k, \nabla^2 g_j(x_k)\hat{d}_k\rangle + O(\|d_k^0\|^\tau), \qquad j \in I(x^*), \end{aligned}$$

since $\tau < 3$. Rearranging to give an expression for $g_j(x_k)$ and then substituting into the third term on the right-hand side of (3.18) for each j gives

$$\begin{aligned} & f(x_k + \hat{d}_k + \tilde{d}_k) - f(x_k) = \\ & \qquad \frac{1}{2}\langle \nabla f(x_k), \hat{d}_k \rangle + \frac{1}{2}\sum_{j=1}^{m} \hat{\lambda}_k^j g_j(x_k) \\ & \qquad + \frac{1}{2}\hat{d}_k^T \left(\nabla^2 f(x_k) + \sum_{j=1}^{m} \hat{\lambda}_k^j \nabla^2 g_j(x_k) - H_k \right) \hat{d}_k \\ & \qquad + O(\|d_k^0\|^\tau). \end{aligned}$$

Subtracting $\alpha\langle \nabla f(x_k), \hat{d}_k \rangle$ from both sides and invoking Lemma 19 shows there exist constants $\nu_2,\ \nu_3 > 0$ such that, since $\tau > 2$,

$$\begin{aligned} & f(x_k + \hat{d}_k + \tilde{d}_k) - f(x_k) - \alpha\langle \nabla f(x_k), \hat{d}_k \rangle \leq \\ & \quad (\frac{1}{2} - \alpha)\langle \nabla f(x_k), \hat{d}_k \rangle + \frac{1}{2}\hat{d}_k^T P_k \left(\nabla^2 f(x_k) + \sum_{j=1}^{m} \hat{\lambda}_k^j \nabla^2 g_j(x_k) - H_k \right) P_k \hat{d}_k \\ & \quad - \left(\nu_2 - \nu_3 \left(\|\hat{d}_k\| + \nu_3 \|g_k\| \right) \left\| \nabla^2 f(x_k) + \sum_{j=1}^{m} \hat{\lambda}_k^j \nabla^2 g_j(x_k) - H_k \right\| \right) \cdot \|g_k\| \\ & \quad + o(\|d_k^0\|^2). \end{aligned}$$

Since $\hat{d}_k \to 0$ and $g_k \to 0$ and all sequences are bounded, the third term on the right-hand side is negative for all k sufficiently large, hence

$$\begin{aligned} & f(x_k + \hat{d}_k + \tilde{d}_k) - f(x_k) - \alpha\langle \nabla f(x_k), \hat{d}_k \rangle \leq \\ & \qquad (\frac{1}{2} - \alpha)\langle \nabla f(x_k), \hat{d}_k \rangle + \frac{1}{2}\hat{d}_k^T P_k \Big(\nabla_{xx}^2 L(x_k, \hat{\lambda}_k) - H_k \Big) P_k \hat{d}_k \\ & \qquad + o(\|d_k^0\|^2). \end{aligned}$$

Assumption 7 says that $P_k(\nabla_{xx}^2 L(x_k, \hat{\lambda}_k) - H_k) P_k \hat{d}_k = o(\|\hat{d}_k\|)$. This, along with

Lemma 19 implies

$$\begin{aligned} f(x_k + \hat{d}_k + \tilde{d}_k) - f(x_k) - \alpha\langle \nabla f(x_k), \hat{d}_k \rangle & \\ &\leq -\nu_1(\frac{1}{2} - \alpha)\|d_k^0\|^2 + o(\|d_k^0\|^2) \\ &\leq 0, \end{aligned}$$

for all k sufficiently large. Thus we have shown that the conditions of the line search in *Step 2* are satisfied with $t_k = 1$ for all k sufficiently large. □

A consequence of Lemmas 17, 18, and 20 is that the algorithm generates a convergent sequence of iterates satisfying

$$x_{k+1} - x_k = d_k^0 + O(\|d_k^0\|^2). \tag{3.19}$$

This allows us to apply, with some modification, the argument used by Powell in [60] to establish a 2-step superlinear rate of convergence, the main result of this section. The modification of Powell's argument to our case is given in the appendix.

Theorem 4. *Algorithm* **FSQP′** *generates a sequence* $\{x_k\}$ *which converges 2-step superlinearly to* x^**, i.e.*

$$\lim_{k\to\infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

3.4 Implementation and Numerical Results

In our implementation of Algorithm **FSQP′** we allow for some classification of the constraints in order to exploit structure. In particular, the implementation contains special provisions for linear (affine) constraints and simple bounds on the variables. The general problem solved is

$$\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & g_j(x) \le 0, && j = 1, \dots, m_n, \\
& \langle a_j, x \rangle + b_j \le 0, && j = 1, \dots, m_a, \\
& x^\ell \le x \le x^u,
\end{aligned}$$

where $a_j \in \mathbb{R}^n$, $b_j \in \mathbb{R}$, $j = 1, \dots, m_a$, and x^ℓ, $x^u \in \mathbb{R}^n$ with $x^\ell < x^u$ (componentwise). The linear constraints and bounds require no "tilting" and may be directly incorporated into $\widehat{QP}(x_k, H_k, \eta_k)$, i.e.

$$\begin{aligned}
\min \quad & \tfrac{1}{2}\langle \hat{d}, H_k \hat{d} \rangle + \hat{\gamma} \\
\text{s.t.} \quad & \langle \nabla f(x_k), \hat{d} \rangle \le \hat{\gamma}, \\
& g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle \le \hat{\gamma} \cdot \eta_k^j, && j = 1, \dots, m_n, \\
& \langle a_j, x_k + \hat{d} \rangle + b_j \le 0, && j = 1, \dots, m_a, \\
& x^\ell - x_k \le \hat{d} \le x^u - x_k.
\end{aligned}$$

Note that a distinct value of η_k is maintained for each nonlinear constraint, i.e η_k^j, $j = 1, \dots, m_n$. This helps significantly in practice while not affecting the analysis. We define the active sets in the implementation as

$$\hat{I}_k^n = \{\ j \mid g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k \rangle - \hat{\gamma}_k \cdot \eta_k^j > -\sqrt{\epsilon_m}\ \}$$

$$\hat{I}_k^a = \{\ j \mid \langle a_j, x_k + \hat{d}_k \rangle + b_j > -\sqrt{\epsilon_m}\ \}$$

where ϵ_m is the machine precision. As before, let $\hat{\lambda}_k^j \in \mathbb{R}^{m_n}$ be the QP multipliers corresponding to the nonlinear constraints. Define $\hat{\lambda}_k^a \in \mathbb{R}^{m_a}$, $\zeta_k^u \in \mathbb{R}^n$, and $\zeta_k^l \in \mathbb{R}^n$ as the QP multipliers corresponding to the affine constraints, the upper bounds, and the lower bounds respectively. The *binding* sets are defined as

$$\hat{I}_k^{b,n} = \{\ j \mid \hat{\lambda}_k^j > 0\ \}, \qquad \hat{I}_k^{b,a} = \{\ j \mid \hat{\lambda}_k^{a,j} > 0\ \},$$

$$\hat{I}_k^{b,l} = \{\ j \mid \zeta_k^{l,j} > 0\ \}, \qquad \hat{I}_k^{b,u} = \{\ j \mid \zeta_k^{u,j} > 0\ \}.$$

Of course, no bending is required from $\tilde{d}_k$ for affine constraints and simple bounds. Hence, if $\hat{I}_k^n = \emptyset$, we simply set $\tilde{d}_k = 0$, otherwise the implementation attempts to compute $\tilde{d}_k$ as the solution of

$$\begin{aligned}
\min \quad & \langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d}) \rangle + \langle \nabla f(x_k), \hat{d}_k + \tilde{d} \rangle \\
\text{s.t.} \quad & g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d} \rangle = -\min\{10^{-2}\|\hat{d}_k\|, \|\hat{d}_k\|^\tau\}, \quad && j \in \hat{I}_k^n, \\
& \langle a_j, x_k + \hat{d}_k + \tilde{d} \rangle + b_j = 0, && j \in \hat{I}_k^a, \\
& \tilde{d}^j = x^u - x_k^j - \hat{d}_k^j, && j \in \hat{I}_k^{b,u}, \\
& \tilde{d}^j = x^l - x_k^j - \hat{d}_k^j, && j \in \hat{I}_k^{b,l}.
\end{aligned}$$

Since not all simple bounds are included in the computation of $\tilde{d}_k$, it is possible that $x_k + \hat{d}_k + \tilde{d}_k$ will not satisfy all bounds. To take care of this, we simply "clip" $\tilde{d}_k$ so that the bounds are satisfied. Specifically, for the upper bounds, we perform the following:

```
**for** $j \notin \hat{I}_k^{b,u}$ **do**
  **if** $(\tilde{d}_k^j \geq x^u - x_k^j - \hat{d}_k^j)$ **then**
    $\tilde{d}_k^j \leftarrow x^u - x_k^j - \hat{d}_k^j$
**end**
```

The same procedure, mutatis mutandis, is executed for the lower bounds. We note that such a procedure has no effect on the convergence analysis of Section 3.3 since, locally, the active set is correctly identified and a full step along $\hat{d}_k + \tilde{d}_k$ is always accepted.

Due to convexity of affine constraints, in the line search of *Step 2* we first generate an upper bound on the step size $\bar{t}_k \leq 1$ using the affine constraints that were not used in the computation of $\tilde{d}_k$. Once these constraints are satisfied, they need not be checked again. Finally, the least squares problem used to compute $\widehat{d_k^0}$ is modified similarly. In the implementation, $\widehat{d_k^0}$ is only computed if $m_n > 0$, in which case we use

$$\begin{array}{lll} \min & \frac{1}{2}\langle \widehat{d^0}, H_k \widehat{d^0} \rangle + \langle \nabla f(x_k), \widehat{d^0} \rangle & \\ \text{s.t.} & g_j(x_k) + \langle \nabla g_j(x_k), \widehat{d^0} \rangle = 0, & j \in \hat{I}_{k-1}^{b,n}, \\ & \langle a_j, x_k + \widehat{d^0} \rangle + b_j = 0, & j \in \hat{I}_{k-1}^{b,a}, \\ & \widehat{d^0}^j = x^u - x_k^j, & j \in \hat{I}_{k-1}^{b,u}, \\ & \widehat{d^0}^j = x^l - x_k^j, & j \in \hat{I}_{k-1}^{b,l}. \end{array}$$

It was mentioned above that, in the implementation, we maintain a separate tilting parameter η_k^j for each nonlinear constraint. In particular, the η_k^j's are different because we use a different scaling C_k^j for each nonlinear constraint. In the algorithm description and in the analysis all that was required of C_k was that it remain bounded and bounded away from zero. In practice, though, performance of the algorithm is critically dependent upon the choice of C_k. For our implementation, an adaptive scheme was chosen in which C_k^j is increased if $g_j(\cdot)$ caused a failure in the line search. Otherwise, if $f(\cdot)$ caused a failure in the line search, C_k is decreased. Specifically, our update rule is as follows,

```
**if** ($g_j(\cdot)$ caused line search failure) **then** $C_{k+1}^j \leftarrow C_k^j \cdot \delta_c$
**else if** ($f(\cdot)$ caused line search failure) **then** $C_{k+1}^j \leftarrow C_k^j / \delta_c$
**if** ($C_{k+1}^j < \underline{C}$) **then** $C_{k+1}^j \leftarrow \underline{C}$
**if** ($C_{k+1}^j > \overline{C}$) **then** $C_{k+1}^j \leftarrow \overline{C}$
```

where $\delta_c > 1$.

Another aspect of the algorithm which was purposefully left vague in Sections 3.2 and 3.3 was the updating scheme for the Hessian estimates H_k. In the implementation, we use the BFGS update with Powell's modification [61]. Specifically, define

$$\delta_{k+1} \triangleq x_{k+1} - x_k$$

$$\gamma_{k+1} \triangleq \nabla_x L(x_{k+1}, \hat{\lambda}_k) - \nabla_x L(x_k, \hat{\lambda}_k),$$

where, in an attempt to better approximate the true multipliers, if $\hat{\mu}_k > \sqrt{\epsilon_m}$ we normalize as follows

$$\hat{\lambda}_k^j \leftarrow \frac{\hat{\lambda}_k^j}{\hat{\mu}_k}, \quad j = 1, \ldots, m_n.$$

A scalar $\theta_{k+1} \in (0, 1]$ is then defined by

$$\theta_{k+1} \triangleq \begin{cases} 1, & \text{if } \delta_{k+1}^T \gamma_{k+1} \geq 0.2 \cdot \delta_{k+1}^T H_k \delta_{k+1}, \\ \dfrac{0.8 \cdot \delta_{k+1}^T H_k \delta_{k+1}}{\delta_{k+1}^T H_k \delta_{k+1} - \delta_{k+1}^T \gamma_{k+1}}, & \text{otherwise.} \end{cases}$$

Defining $\xi_{k+1} \in \mathbb{R}^n$ as

$$\xi_{k+1} \triangleq \theta_{k+1} \cdot \gamma_{k+1} + (1 - \theta_{k+1}) \cdot H_k \delta_{k+1},$$

the rank two Hessian update is

$$H_{k+1} = H_k - \frac{H_k \delta_{k+1} \delta_{k+1}^T H_k}{\delta_{k+1}^T H_k \delta_{k+1}} + \frac{\xi_{k+1} \xi_{k+1}^T}{\delta_{k+1}^T \xi_{k+1}}.$$

Note that while it is not clear whether the resultant sequence $\{H_k\}$ will, in fact, satisfy Assumption 7, this update scheme is known to perform very well in practice.

Our implementation calls the Goldfarb-Idnani based active set QP solver QLD due to Powell and Schittkowski [70]. QLD uses dense linear algebra and does not allow "warm starts", i.e. does not allow the user to supply an initial guess for the QP multipliers. For simplicity, we not only used QLD to solve $\widehat{QP}(x_k, H_k, \eta_k)$, but also the least squares problems. Of course, this was likely not too inefficient since the active set is known automatically for these problems. In order to guarantee that the algorithm terminates after a finite number of iterations with an approximate solution, the stopping criterion of *Step 1(ii)* is changed to

$$\textbf{if} \ \ (\|\hat{d}_k\| \leq \epsilon) \ \ \textbf{stop},$$

where $\epsilon > 0$ is small. Finally, note that during the line search of *Step 2*, as soon as it is determined that the given trial point does not satisfy the descent criterion or a particular constraint, no more constraints are evaluated. In this case, a new trial point is immediately computed and the trial evaluations start over from the beginning. In order to reduce the number of constraint function evaluations, the constraint which caused the failure is always checked first at the new trial point, as it is most likely to be infeasible.

In order to test the implementation, we selected several problems from [28] which provided feasible initial points and contained no equality constraints. The results are reported in Table 3.1. For all problems we used the parameter values

$$\begin{array}{lll} \alpha = 0.1, & \beta = 0.5, & \tau = 2.5, \\ \epsilon_\ell = \min\{1, \sqrt{\epsilon}\}, & \underline{C} = 1 \times 10^{-3}, & \overline{C} = 1 \times 10^{3}, \\ \delta_c = 10, & \bar{D} = 10 \cdot \epsilon_\ell. & \end{array}$$

Further, we always set $H_0 = I$ and $\eta_0^j = 1 \times 10^{-2}$, $C_0^j = 1$, $j = 1, \ldots, m_n$.

In Table 3.1 we compare our implementation with CFSQP [36], the implementation of Algorithm **FSQP** as described in [51]. The column labeled # lists the problem number as given in [28], the column labeled `ALGO` tells which algorithm was used to solve the given problem (the names are self-explanatory). The next three columns give the size of the problem following the conventions of this section. The columns labeled `NF`, `NG`, and `IT` give the number of objective function evaluations, nonlinear constraint function evaluations, and iterations required to solve the problem, respectively. Finally, $f(x^*)$ is the objective function value at the final iterate and ϵ is the tolerance for the size of the search direction (the stopping criterion). The value of ϵ was chosen in order to obtain approximately the same precision as reported in [28] for each problem.

The results reported in Table 3.1 are very encouraging. The performance of our implementation of Algorithm **FSQP′** is essentially identical to that of CFSQP (Algorithm **FSQP**). Of course, Algorithm **FSQP′** requires substantially less work per iteration than Algorithm **FSQP**. Thus, in the case that the work to generate a new iterate dominates the work to evaluate the objectives and constraints, the new algorithm is at a clear advantage.

3.5 Modification of Powell's Argument

In this appendix we discuss how the arguments given by Powell in Sections 2 and 3 of [60] may be used, with some modification, to prove Theorem 4. To avoid confusion, we will refer to lemmas from [60] as Lemma P.n, where n is the number as it appears in [60]. We begin by noting that all of Powell's assumptions outlined at the beginning of Section 2 in [60] hold in our case (under the strengthened assumptions of Section 3.3.2). Also, Lemmas P.1 and P.2 are already established by our Lemmas 14 and 16. These Lemmas show that the active set is exactly identified by the QP multipliers for all k sufficiently large. In view of this, and since Lemma 20 shows that $t_k = 1$ for all k sufficiently large, the inactive constraints eventually have no effect on the computation of a new iterate. Thus, without loss of generality, it may be assumed here that we are generating iterates converging to a solution of the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) = 0, \qquad j \in I(x^*), \end{aligned} \tag{P^+}$$

Let $L^+ : \mathbb{R}^n \times \mathbb{R}^{|I(x^*)|} \to \mathbb{R}$ be the corresponding Lagrangian function and, recalling our notation introduced in Section 3.3.2, let λ^{*+} be the optimal multiplier for (P^+).

Lemma P.3, which establishes that the SQP direction d_k^0 is unchanged when the matrix H_k is perturbed by a symmetric matrix whose kernel includes the orthogonal complement of the constraint gradients, is algorithm independent, hence automatically holds. Following Powell's notation, define

$$h_k \overset{\Delta}{=} P_k \nabla f(x_k),$$

and interpret the symbol "$\sim$" as meaning the ratio of the expression on the left-hand side to the right-hand side is both bounded above and bounded away from zero, as $k \to \infty$. Using the same argument as in Lemma P.4, we can show (recall the definition of g_k from Section 3.3.2)

$$\|d_k^0\| \sim \|g_k\| + \|h_k\|.$$

In view of (3.19), this implies Lemma P.4 still holds in our case.

Unfortunately, the proof of Lemma P.5 will not work in our context. Thus, we establish this result here.

Lemma 21. $\|x_k - x^*\| \sim \|g_k\| + \|h_k\|$.

Proof. We begin by showing that $\nabla^2 L^+(x^*, \lambda^{*+})$ (by which we mean the second derivative with respect to both x and λ) is non-singular. Let $R^* \overset{\Delta}{=} \lim_{k\to\infty} R_k$. Suppose there exists $z = (y^T, u^T)^T \in \mathbb{R}^{n+|I(x^*)|}$ such that $\nabla^2 L^+(x^*, \lambda^*)z = 0$. Then, using complementary slackness we can substitute $\nabla^2_{xx} L(x^*, \lambda^*)$ for $\nabla^2_{xx} L^+(x^*, \lambda^{*+})$, obtaining

$$\begin{bmatrix} \nabla^2_{xx} L(x^*, \lambda^*) & R^* \\ R^{*T} & 0 \end{bmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = 0.$$

So, $R^{*T} y = 0$ and $y^T \nabla^2_{xx} L(x^*, \lambda^*) y = -(R^{*T} y)^T u = 0$, which, in view of Assumption 5, implies $y = 0$. This, in turn, implies $R^* u = 0$, which, by Assumption 3 requires $u = 0$. Thus, we have shown that $\nabla^2 L^+(x^*, \lambda^{*+})$ is non-singular.

Note that we may write

$$\begin{aligned} &\nabla L^+(x_k, \lambda_k^{0+}) \\ &\quad = \int_0^1 \nabla^2 L^+(x^* + t(x_k - x^*), \lambda^{*+} + t(\lambda_k^{0+} - \lambda^{*+})) \begin{pmatrix} x_k - x^* \\ \lambda_k^{0+} - \lambda^{*+} \end{pmatrix} dt \\ &\quad \overset{\Delta}{=} \overline{D}_k \begin{pmatrix} x_k - x^* \\ \lambda_k^{0+} - \lambda^{*+} \end{pmatrix}. \end{aligned}$$

Since $x_k \to x^*$ and $\lambda_k^{0+} \to \lambda^{*+}$, it follows from our regularity Assumption $2'$ that $\overline{D}_k \to \nabla^2 L^+(x^*, \lambda^{*+})$. Non-singularity of $\nabla^2 L^+(x^*, \lambda^{*+})$ implies that for all k sufficiently large, $\overline{D}_k$ is non-singular and there exists $\overline{M} > 0$ such that

$$\|\overline{D}_k^{-1}\| \leq \overline{M},$$

for k large enough. Thus,

$$\begin{aligned} \|x_k - x^*\| &\leq \left(\|x_k - x^*\|^2 + \|\lambda_k^{0+} - \lambda^{*+}\|^2\right)^{\frac{1}{2}} \\ &= \left\|\overline{D}_k^{-1} \nabla L^+(x_k, \lambda_k^{0+})\right\|, \end{aligned}$$

(where we are using the Euclidean norm) which implies

$$\|x_k - x^*\| \leq \overline{M} \left\| \nabla L^+(x_k, \lambda_k^{0+}) \right\|, \tag{3.20}$$

for all k sufficiently large (note that we are using the Euclidean norm). Recall now that, for k large enough, the SQP direction satisfies

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^0 \\ \lambda_k^{0+} \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix}.$$

This can be solved for d_k^0, yielding

$$\begin{aligned} d_k^0 &= -H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} g_k \\ &\quad - \left[H_k^{-1} - H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} R_k^T H_k^{-1} \right] \nabla f(x_k) \\ &= -H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} g_k \\ &\quad - \left[H_k^{-1} - H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} R_k^T H_k^{-1} \right] \\ &\qquad \cdot (P_k + R_k (R_k^T R_k)^{-1} R_k^T) \nabla f(x_k) \\ &= -H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} g_k \\ &\quad - \left[H_k^{-1} - H_k^{-1} R_k (R_k^T H_k^{-1} R_k)^{-1} R_k^T H_k^{-1} \right] P_k \nabla f(x_k) \\ &\triangleq B_k g_k + E_k h_k, \end{aligned}$$

where B_k and E_k are bounded for large k, and we have used the trivial identity $P_k + R_k (R_k^T R_k)^{-1} R_k^T = I$. Now, in view of the optimality conditions (3.4),

$$\begin{aligned} \nabla_x L^+(x_k, \lambda_k^{0+}) &= -H_k d_k^0 \\ &= -H_k B_k g_k - H_k E_k h_k. \end{aligned}$$

Thus, there exist $K_1,\ K_2 > 0$ such that for large k

$$\|\nabla_x L^+(x_k, \lambda_k^{0+})\| \leq K_1 \|g_k\| + K_2 \|h_k\|. \tag{3.21}$$

Finally, since $\nabla_\lambda L^+(x_k, \lambda_k^{0+}) = g_k$, we conclude from (3.20) and (3.21) that there exists $K_3 > 0$ such that for large k

$$\|x_k - x^*\| \leq K_3 \cdot (\|g_k\| + \|h_k\|) \, .$$

To go the other direction, expanding $g(\cdot)$ about x^* (recall that for this argument $g : \mathbb{R}^n \to \mathbb{R}^{|I(x^*)|}$) and noting that $P_k \nabla g_j(x_k) = 0$ for all k, we have

$$\begin{aligned}
\|g_k\| + \|h_k\| &= \|g(x^*) + R_k^T(x_k - x^*) + O(\|x_k - x^*\|^2)\| \\
&\qquad + \|P_k \nabla_x L^+(x_k, \lambda^{*+})\| \\
&= \|R_k^T(x_k - x^*) + O(\|x_k - x^*\|^2)\| \\
&\qquad + \|P_k(\nabla_x L^+(x^*, \lambda^{*+}) + \nabla^2_{xx} L^+(x^*, \lambda^{*+})(x_k - x^*) \\
&\qquad\quad + O(\|x_k - x^*\|^2)\| \\
&= \|R_k^T(x_k - x^*)\| + \|P_k \nabla^2_{xx} L^+(x^*, \lambda^{*+})(x_k - x^*)\| \\
&\qquad\quad + O(\|x_k - x^*\|^2) \\
&\leq K_4 \|x_k - x^*\| + O(\|x_k - x^*\|^2),
\end{aligned}$$

for some constant $K_4 > 0$, and the result follows. □

Lemma P.6 requires some additional explanation in our case. In particular, we need to justify/modify equations (3.3), (3.8), and (3.9) in [60]. To begin with, consider for all k sufficiently large (and recall that we are only interested in $j \in I(x^*)$ here)

$$\begin{aligned}
g_j(x_{k+1}) &= g_j(x_k + d_k^0 + O(\|d_k^0\|^2) \\
&= g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle + O(\|d_k^0\|^2) \\
&= O(\|x_{k+1} - x_k\|^2).
\end{aligned}$$

Thus equation (3.3) holds. If $O(\|x_{k+1} - x_k\|^2)$ is added to the right hand side of equation (3.8), and to both sides of equation (3.9), then the same argument holds for the sequences generated by Algorithm **FSQP′**. Finally, Theorem P.1 is the same as our Theorem 4 and the argument used in [60] may be used to prove Theorem 4.

#	ALGO	n	m_a	m_n	NF	NG	IT	$f(x^*)$	ϵ
12	NEW	2	0	1	7	14	7	-3.0000000e+1	1e-6
	CFSQP				7	14	7	-3.0000000e+1	
29	NEW	3	0	1	11	20	10	-2.2627417e+1	1e-5
	CFSQP				11	20	10	-2.2627417e+1	
30	NEW	3	0	1	18	35	18	1.0000000e+0	1e-7
	CFSQP				18	35	18	1.0000000e+0	
31	NEW	3	0	1	9	25	8	6.0000000e+0	1e-5
	CFSQP				9	19	7	6.0000000e+0	
33	NEW	3	0	2	4	11	4	-4.0000000e+0	1e-8
	CFSQP				4	11	4	-4.0000000e+0	
34	NEW	3	0	2	8	32	8	-8.3403245e-1	1e-8
	CFSQP				7	28	7	-8.3403244e-1	
43	NEW	4	0	3	9	45	8	-4.4000000e+1	1e-5
	CFSQP				10	46	8	-4.4000000e+1	
66	NEW	3	0	2	8	30	8	5.1816327e-1	1e–8
	CFSQP				8	30	8	5.1816327e-1	
84	NEW	5	0	6	4	32	4	-5.2803351e+6	1e-8
	CFSQP				4	30	4	-5.2803351e+6	
93	NEW	6	0	2	14	55	12	1.3507596e+2	1e-5
	CFSQP				16	62	13	1.3507596e+2	
113	NEW	10	3	5	13	116	13	2.4306210e+1	1e-3
	CFSQP				12	108	12	2.4306377e+1	
117	NEW	15	0	5	19	179	17	3.2348679e+1	1e-4
	CFSQP				20	219	19	3.2348679e+1	

Table 3.1: Numerical results for **FSQP′**.

Chapter 4

Mini-Max Algorithm

4.1 Introduction

In this chapter, we extend the basic algorithm of Chapter 3 to solve the constrained mini-max problem

$$\begin{aligned} \min \quad & F(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{aligned} \tag{M}$$

where

$$F(x) \overset{\Delta}{=} \max\{\ f_j(x) \mid j = 1, \ldots, p\ \},$$

and the functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. Of course, since $F(x)$ is a non-differentiable function, (M) is a non-smooth optimization problem. As a consequence, Algorithm **FSQP′** may not be applied directly to solve (M).

It is well-known that (M) may be transformed into an equivalent *smooth* nonlinear programming problem by adding a variable as follows

$$\begin{aligned} \min_{(x,\gamma)} \quad & \gamma \\ \text{s.t.} \quad & f_j(x) \le \gamma, \qquad j = 1, \dots, p, \\ & g_j(x) \le 0, \qquad j = 1, \dots, m. \end{aligned} \tag{M'}$$

It is not difficult to show that $x^* \in \mathbb{R}^n$ is a local minimizer for (M) if, and only if, $(x^*, F(x^*)) \in \mathbb{R}^n \times \mathbb{R}$ is a local minimizer for (M'). Thus, we could apply Algorithm **FSQP′** to (M') in order to solve the non-smooth problem (M). It turns out, though, that there are a few reasons why this may not be a desirable approach. To begin with, blindly applying a standard nonlinear programming algorithm to (M') ignores a great deal of structure which could be exploited in (M). Further, in the context of feasible direction algorithms, there is no reason why any additional effort should be expended maintaining "feasibility" for the objective functions which appear as constraints in (M'). Finally, if the algorithm of Chapter 3 were to be applied to (M'), the line search would enforce descent on γ for each iteration. As it is possible that, at any particular iteration, none of the constraints $f_j(x) \le \gamma$ will be active[1], the generated sequence of iterates will *not* be guaranteed to exhibit the objective function descent property

$$F(x_{k+1}) < F(x_k),$$

which is useful in many applications. It is true that only a simple modification of **FSQP′** would be required to ensure the descent property does hold for a mini-max problem when posed in the form (M'). Still, this largely ignores the structure of the problem and is not the most efficient way to proceed.

[1] Though, at least one will be active at the solution.

A number of authors have considered the problem (M). The unconstrained problem was first considered by Han in [24, 23]. Han's approach was essentially to apply the SQP algorithm of [22] to the equivalent problem (M') while carefully exploiting the special structure. Polak, Mayne, and Higgins [56, 57] successively solve quadratic approximations of the original problem, using exact second derivatives, in order to obtain search directions. Their algorithms achieve quadratic convergence under fairly strict conditions. In [77], Zhou and Tits proposed a new SQP-based algorithm for the unconstrained mini-max problem which incorporated a nonmonotone line search scheme in an attempt to avoid the Maratos effect without computing a second order correction. The same authors proposed an algorithm based on a monotone line search for mini-max problems with a large number of objective functions in [78]. The constrained problem has been considered by, e.g., Kiwiel in [32], Panier and Tits in [47], and Zhou in [76], all in the context of feasible iterates. In [76], Zhou extends the nonmonotone line search-based algorithm of [77] to handle the constrained case. The algorithm of [47] extends the feasible SQP algorithm of [48] to handle mini-max objective functions. A recent algorithm for the constrained mini-max problem which does not generate feasible iterates is the augmented Lagrangian approach of Rustem and Nguyen [69]. The extension of Algorithm **FSQP′** discussed in this chapter was inspired by the algorithms of [47, 76].

In Section 4.2, we present the details of our extension of **FSQP′**. In Section 4.3, we show that the convergence results of Section 3.3 are preserved, i.e. the algorithm is globally convergent, as well as locally superlinearly convergent. The implementation of Algorithm **FSQP′** has been extended to solve (M) and we show in Section 4.4 that the numerical results are, again, quite promising.

4.2 Algorithm

We begin by making a few definitions. Let $I \stackrel{\Delta}{=} \{1, \dots, m\}$ and $J \stackrel{\Delta}{=} \{1, \dots, p\}$. As before, let X denote the feasible set for (M), i.e.

$$X \stackrel{\Delta}{=} \{\ x \in \mathbb{R}^n \mid g_j(x) \leq 0, \quad j \in I\ \}.$$

Given $x \in X$, let

$$I(x) \stackrel{\Delta}{=} \{\ j \in I \mid g_j(x) = 0\ \}$$

denote the set of active constraints at x, and let

$$J(x) \stackrel{\Delta}{=} \{\ j \in J \mid f_j(x) = F(x)\ \}$$

denote the set of active objective functions at x. The following assumptions will hold throughout this chapter.

Assumption 1: The set X is non-empty.

Assumption 2: The functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j \in J$, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j \in I$, are continuously differentiable.

Assumption 3: For all $x \in X$, the set $\{\nabla g_j(x) \mid j \in I(x)\}$ is linearly independent. Further, for all $x \in X$ with $I(x) \neq \emptyset$, the set $\{\nabla f_j(x) \mid j \in J(x)\}$ is linearly independent.

Recall from Section 2.1, a point $x \in \mathbb{R}^n$ is said to be a *Karush-Kuhn-Tucker (KKT)* point for the problem (M) if there exist scalars (*KKT multipliers*) μ^j,

$j \in J$, and λ^j, $j \in I$, satisfying

$$\begin{cases} \displaystyle\sum_{j \in J} \mu^j \nabla f_j(x) + \sum_{j \in I} \lambda^j \nabla g_j(x) = 0, \\ \displaystyle\sum_{j \in J} \mu^j = 1, \\ g_j(x) \leq 0, \quad j \in I, \\ \mu^j \left(f_j(x) - F(x) \right) = 0 \text{ and } \mu^j \geq 0, \quad j \in J, \\ \lambda^j g_j(x) = 0 \text{ and } \lambda^j \geq 0, \quad j \in I. \end{cases} \tag{4.1}$$

The Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ for (M) is defined by

$$L(x, \mu, \lambda) \triangleq \sum_{j \in J} \mu^j f_j(x) + \sum_{j \in I} \lambda^j g_j(x).$$

Given $x \in \mathbb{R}^n$ and $J' \subseteq J$, define

$$F_{J'}(x) \triangleq \max\{ \ f_j(x) \mid j \in J' \ \}.$$

Further, given a direction $d \in \mathbb{R}^n$, let

$$F'(x, d) \triangleq \max\{ \ f_j(x) + \langle \nabla f_j(x), d \rangle \mid j \in J \ \} - F(x),$$

i.e. a first-order approximation of $F(x + d) - F(x)$. Loosely speaking, we extend Algorithm **FSQP′** to solve the non-smooth problem (M) by replacing all instances of the directional derivative of the objective function with $F'(x, d)$. In particular, given $x \in X$, $0 < H = H^T \in \mathbb{R}^{n \times n}$, and $\eta \geq 0$, let

$$(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta), \hat{\gamma}(x, H, \eta)) \in \mathbb{R}^n \times \mathbb{R}$$

be the unique solution of the QP

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle \hat{d}, H\hat{d} \rangle + \hat{\gamma} \\ \text{s.t.} \quad & F'(x, \hat{d}) \leq \hat{\gamma}, \\ & g_j(x) + \langle \nabla g_j(x), \hat{d} \rangle \leq \hat{\gamma} \cdot \eta, \quad j = 1, \ldots, m. \end{aligned} \qquad \widehat{QP}(x, H, \eta)$$

To see that this is, indeed, a QP, note that it may be equivalently written as

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle \hat{d}, H\hat{d}\rangle + \hat{\gamma} \\ \text{s.t.} \quad & f_j(x) + \langle \nabla f_j(x), \hat{d}\rangle \leq F(x) + \hat{\gamma}, \quad j = 1, \ldots, p, \\ & g_j(x) + \langle \nabla g_j(x), \hat{d}\rangle \leq \hat{\gamma} \cdot \eta, \qquad\quad j = 1, \ldots, m. \end{aligned}$$

As in Chapter 3, we will make frequent use of the optimality conditions for $\widehat{QP}(x, H, \eta)$, which are readily derived from the second form given above. Specifically, $(\hat{d}, \hat{\gamma})$ is a KKT point for $\widehat{QP}(x, H, \eta)$ if there exist multipliers $\hat{\mu} \in \mathbb{R}^p$ and $\hat{\lambda} \in \mathbb{R}^m$ which, together with $(\hat{d}, \hat{\gamma})$, satisfy

$$\begin{cases} \begin{bmatrix} H\hat{d} \\ 1 \end{bmatrix} + \sum_{j \in J} \hat{\mu}^j \begin{bmatrix} \nabla f_j(x) \\ -1 \end{bmatrix} + \sum_{j \in I} \hat{\lambda}^j \begin{bmatrix} \nabla g_j(x) \\ -\eta \end{bmatrix} = 0, \\ f_j(x) + \langle \nabla f_j(x), \hat{d}\rangle \leq F(x) + \hat{\gamma} \cdot \eta, \quad \forall j \in J, \\ g_j(x) + \langle \nabla g_j(x), \hat{d}\rangle \leq \hat{\gamma} \cdot \eta, \quad \forall j \in I, \\ \hat{\mu}^j \left(f_j(x) + \langle \nabla f(x), \hat{d}\rangle - F(x) - \hat{\gamma} \right) = 0 \text{ and } \hat{\mu}^j \geq 0, \quad \forall j \in J \\ \hat{\lambda}^j \left(g_j(x) + \langle \nabla g_j(x), \hat{d}\rangle - \hat{\gamma} \cdot \eta \right) = 0 \text{ and } \hat{\lambda}^j \geq 0, \quad \forall j \in I. \end{cases} \tag{4.2}$$

A simple consequence of the first equation in (4.2), which will be used throughout our analysis, is an affine relationship amongst the multipliers, i.e.

$$\sum_{j \in J} \hat{\mu}^j + \eta \cdot \sum_{j=1}^{m} \hat{\lambda}^j = 1. \tag{4.3}$$

At iteration k, in order to update the tilting parameter η_k, we will again have to estimate the SQP direction $d^0(x_k, H_k)$, which, for mini-max problems, is defined via the following QP (see, e.g., [47])

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle d^0, H_k d^0\rangle + F'(x_k, d^0) \\ \text{s.t.} \quad & g_j(x_k) + \langle \nabla g_j(x_k), d^0\rangle \leq 0, \quad j \in I. \end{aligned} \qquad QP^0(x_k, H_k)$$

Of course, in the interest of reducing computational cost per iteration, we would again like to approximate $d^0(x_k, H_k)$ by instead solving an *equality* constrained QP. Given $J_k \subseteq J$ and $I_k \subseteq I$ let

$$(\widehat{d^0}, \widehat{\gamma^0}) = (\widehat{d^0}(x_k, H_k, J_k, I_k), \widehat{\gamma^0}(x_k, H_k, J_k, I_k)) \in \mathbb{R}^n \times \mathbb{R}$$

be the solution, if it exists, of the equality constrained QP

$$\begin{array}{lllr} \min & \frac{1}{2}\langle \widehat{d^0}, H_k \widehat{d^0} \rangle + \widehat{\gamma^0} & & \\ \text{s.t.} & f_j(x_k) + \langle \nabla f_j(x_k), \widehat{d^0} \rangle = F(x_k) + \widehat{\gamma^0} & j \in J_k & \qquad LS^0(x_k, H_k, J_k, I_k) \\ & g_j(x_k) + \langle \nabla g_j(x_k), \widehat{d^0} \rangle = 0, & j \in I_k, & \end{array}$$

Note that while this QP is no longer equivalent to a least squares problem (due to the lack of quadratic dependence on $\widehat{\gamma^0}$ in the objective), its solution still only requires the solution of one linear system of equations in $n + |J_k| + |I_k|$ variables. Define

$$\hat{J}_k \triangleq \{\ j \in J \mid f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k \rangle = F(x_k) + \hat{\gamma}_k\ \}$$
$$\hat{I}_k \triangleq \{\ j \in I \mid g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k \rangle = \hat{\gamma}_k \cdot \eta_k\ \}$$

as the active sets from $\widehat{QP}(x_k, H_k, \eta_k)$. It will be shown in Section 4.3.2 that, in order to guarantee superlinear convergence, it is sufficient to choose

$$J_k = \hat{J}_{k-1}, \qquad I_k = \hat{I}_{k-1}.$$

Before we accept $\widehat{d^0_k} = \widehat{d^0}(x_k, H_k, \hat{J}_{k-1}, \hat{I}_{k-1})$ as a "good" estimate of $d^0_k = d^0(x_k, H_k)$ and use it to compute the tilting parameter η_k, we first check that it satisfies a few important conditions. Let $\widehat{\mu^0_k} \in \mathbb{R}^{|\hat{J}_{k-1}|}$ and $\widehat{\lambda^0_k} \in \mathbb{R}^{|\hat{I}_{k-1}|}$ be the multipliers from $LS^0(x_k, H_k, \hat{J}_{k-1}, \hat{I}_{k-1})$, and suppose $\bar{D} > 0$ is some pre-specified number. If $\widehat{d^0_k}$ is to be considered a reasonable estimate of d^0_k, the following conditions must be satisfied:

1. $\widehat{d^0_k}$ exists,

2. $\|\widehat{d^0_k}\| \leq \bar{D}$,

3. $\widehat{\mu^0_k} \geq 0$,

4. $\widehat{\lambda^0_k} \geq 0$,

5. $F_{\hat{J}_{k-1}}(x_k) = F(x_k)$.

The need for the first conditions is obvious. As we know that the estimate is likely only valid in a neighborhood of the solution, and since $d^0(x, H)$ is zero at the solution and small in a neighborhood, we reject overly large estimates. Since the multipliers at the solution are known to be positive, the third and fourth conditions further help to eliminate poor estimates. Finally, the last condition helps to eliminate cases when $\hat{J}_{k-1}$ is a poor estimate of the true active set. We will show in Section 4.3.2 that these conditions will always hold for k large enough. During the early iterations, when the conditions do not hold, it is sufficient to use $\min\{\epsilon_\ell, \|\hat{d}_{k-1}\|^2\}$ in place of $\|\widehat{d^0_k}\|^2$ for the tilting parameter update, where $\epsilon_\ell > 0$ is a small parameter.

Finally, we will again use a second-order correction to avoid the Maratos effect. In [47], the authors suggest the linear least squares problem

$$\begin{array}{lll}\min & \frac{1}{2}\|\tilde{d}\|^2 & \\ \text{s.t.} & f_i(x_k) + \langle \nabla f_i(x_k), \tilde{d}\ \rangle = f_j(x_k) + \langle \nabla f_j(x_k), \tilde{d}\ \rangle & i, j \in J_k, \\ & g_j(x_k) + \langle \nabla g_j(x_k), d \rangle = 0, & j \in I_k,\end{array}$$

where J_k and I_k are the index sets of active objectives and constraints, respectively, from the computation of the search direction. While such an approach would be sufficient to guarantee a step of one is eventually always accepted for our algorithm, we use an equality constrained QP which has fewer constraints and tends to work better in practice (and, of course, also guarantees a step of one is eventually always accepted). At iteration k, let

$$(\tilde{d}_k, \tilde{\gamma}_k) = (\tilde{d}(x_k, \hat{d}_k, H_k, \hat{J}_k, \hat{I}_k), \tilde{\gamma}(x_k, \hat{d}_k, H_k, \hat{J}_k, \hat{I}_k)) \in \mathbb{R}^n \times \mathbb{R}$$

be the solution, if it exists, of

$$\begin{array}{lrl} \min & \multicolumn{2}{l}{\frac{1}{2}\langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d})\rangle + \tilde{\gamma}} \\ \text{s.t.} & f_j(x_k + \hat{d}_k) + \langle \nabla f_j(x_k), \tilde{d}\rangle = F_{\hat{J}_k}(x_k + \hat{d}_k) + \tilde{\gamma}, & j \in \hat{J}_k, \\ & g_j(x_k + \hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d}\rangle = -\|\hat{d}_k\|^\tau, & j \in \hat{I}_k, \\ & & \widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{J}_k, \hat{I}_k) \end{array}$$

where $\tau \in (2, 3)$. We are now in a position to state the complete algorithm.

Algorithm FSQP′-MM

Parameters: $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $\tau \in (2, 3)$, $\epsilon_\ell > 0$, $0 < \underline{C} \leq \overline{C}$, $\bar{D} > 0$.

Data: $x_0 \in X$, $0 < H_0 = H_0^T \in \mathbb{R}^{n \times n}$, $0 < \eta_0 \in \mathbb{R}$.

Step 0 - Initialization. **set** $k \leftarrow 0$.

Step 1 - Computation of search arc.

(i). **compute** $(\hat{d}_k, \hat{\gamma}_k) = (\hat{d}(x_k, H_k, \eta_k), \hat{\gamma}(x_k, H_k, \eta_k))$, the active sets $\hat{J}_k$ and $\hat{I}_k$, and the associated multipliers $\hat{\mu}_k \in \mathbb{R}^p$, $\hat{\lambda}_k \in \mathbb{R}^m$.

(ii). **if** $(\hat{d}_k = 0)$ **then stop**.

(iii). **compute** $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{J}_k, \hat{I}_k)$ if it exists and satisfies $\|\tilde{d}_k\| \leq \|\hat{d}_k\|$. Otherwise, **set** $\tilde{d}_k = 0$.

Step 2 - Arc search. **compute** t_k, the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$F(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq F(x_k) + \alpha t F'(x_k, \hat{d}_k),$$

$$g_j(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq 0, \quad j = 1, \dots, m.$$

Step 3 - Updates.

(i). **set** $x_{k+1} \leftarrow x_k + t_k\hat{d}_k + t_k^2\tilde{d}_k$.

(ii). **compute** a new symmetric positive definite estimate H_{k+1} to the Hessian of the Lagrangian.

(iii). **select** $C_{k+1} \in [\underline{C}, \overline{C}]$.

- $*$ **if** $(\|\hat{d}_k\| < \epsilon_\ell)$ **then**
    - $\cdot$ **compute**, if possible,[2] $\widehat{d^0_{k+1}} = \widehat{d^0}(x_{k+1}, H_{k+1}, \hat{J}_k, \hat{I}_k)$, and the associated multipliers $\widehat{\mu^0_{k+1}} \in \mathbb{R}^{|\hat{J}_k|}$ and $\widehat{\lambda^0_{k+1}} \in \mathbb{R}^{|\hat{I}_k|}$.
    - $\cdot$ **if** $\Big(\widehat{d^0_{k+1}}$ exists and $\|\widehat{d^0_{k+1}}\| \leq \bar{D}$ and $F_{\hat{J}_k}(x_{k+1}) = F(x_{k+1})$ and $\widehat{\mu^0_{k+1}} \geq 0$ and $\widehat{\lambda^0_{k+1}} \geq 0\Big)$ **then set**

      $$\eta_{k+1} \leftarrow C_{k+1} \cdot \|\widehat{d^0_{k+1}}\|^2.$$

    - $\cdot$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \|\hat{d}_k\|^2$.
- $*$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \epsilon_\ell^2$.

(iv). **set** $k \leftarrow k + 1$ and **goto** *Step 1.*

[2] That is, if $LS^0(x_{k+1}, H_{k+1}, \hat{J}_k, \hat{I}_k)$ is non-degenerate.

4.3 Convergence Analysis

In order to establish global convergence to a KKT point for (M), the analysis will exactly follow that of Section 3.3.1. While the chain of results are identical, there are a few critical areas in which the analysis itself differs significantly. We will omit the details of any argument which is a trivial modification of the corresponding argument in Section 3.3.1. The local convergence analysis will require a good bit more effort, though it will also follow the outline of that for Algorithm **FSQP′** in Section 3.3.2.

4.3.1 Global Convergence

The goal of this section will be to show that Algorithm **FSQP′-MM** generates a sequence of iterates $\{x_k\}$ for which all accumulation points are KKT points for (M).

Lemma 22. *Given $H = H^T > 0$, $x \in X$, and $\eta \geq 0$, $\hat{d}(x, H, \eta)$ is well-defined and $(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta), \hat{\gamma}(x, H, \eta))$ is the unique KKT point of $\widehat{QP}(x, H, \eta)$. Furthermore, suppose $\{x_k\}_{k\in\mathbb{N}} \subset X$ is bounded, $\{H_k\}_{k\in\mathbb{N}}$ is bounded away from singularity, and $\{\eta_k\}_{k\in\mathbb{N}} \subset [0, \infty)$. Then $\{\hat{d}(x_k, H_k, \eta_k)\}_{k\in\mathbb{N}}$ is bounded.*

Proof. First note that the feasible set for $\widehat{QP}(x, H, \eta)$ is non-empty, since $(\hat{d}, \hat{\gamma}) = (0, 0)$ is always feasible. The case $\eta > 0$ is similar to that in Lemma 1 and we omit it here. Consider now the case $\eta = 0$. In this case, $(\hat{d}, \hat{\gamma})$ solves $\widehat{QP}(x, H, 0)$ if, and only if, $\hat{d}$ solves $QP^0(x, H)$ and $\hat{\gamma} = F'(x, \hat{d})$. Since $H = H^T > 0$, the objective in $QP^0(x, H)$ is strictly convex and radially unbounded. Further, the feasible set is convex, therefore the solution of $QP^0(x, H)$ is well-defined and unique. This, in turn, implies $(\hat{d}(x, H, 0), \hat{\gamma}(x, H, 0))$ is well-defined and the unique KKT point of the convex problem $\widehat{QP}(x, H, 0)$. For the third claim, note that since $\{H_k\}_{k\in\mathbb{N}}$ is bounded away from singularity and $H_k = H_k^T > 0$, for all k, there exists $\sigma_1 > 0$ such that

$$\langle \hat{d}_k, H_k \hat{d}_k \rangle \geq \sigma_1 \|\hat{d}_k\|^2, \quad \forall k.$$

Let $\hat{j}(k)$ be such that $f_{\hat{j}(k)}(x_k) = F(x_k)$ for all k. As the optimal value of $\widehat{QP}(x_k, H_k, \eta_k)$ is non-positive (since $(0, 0)$ is always feasible),

$$\hat{\gamma}_k \leq -\frac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle,$$

for all k. In view of QP constraint $\hat{j}(k)$,

$$\begin{array}{rcl} \langle \nabla f_{\hat{j}(k)}(x_k), \hat{d}_k \rangle & \leq & -\frac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle \\ & \leq & -\frac{\sigma_1}{2}\|\hat{d}_k\|^2, \end{array}$$

for all k. It immediately follows that

$$\|\hat{d}_k\| \leq \frac{2}{\sigma_1}\|\nabla f_{\hat{j}(k)}(x_k)\|.$$

In view of Assumption 2 and boundedness of $\{x_k\}_{k\in\mathbb{N}}$, $\{\nabla f_j(x_k)\}_{k\in\mathbb{N}}$ is bounded for each j. As there are only finitely many j, $\{\hat{d}_k\}_{k\in\mathbb{N}}$ is bounded. □

Lemma 23. *Given $H = H^T > 0$ and $\eta \geq 0$*

(i). $\hat{\gamma}(x, H, \eta) \leq 0$ for all $x \in X$. Moreover, $\hat{\gamma}(x, H, \eta) = 0$ if, and only if, $\hat{d}(x, H, \eta) = 0$.

(ii). $\hat{d}(x, H, \eta) = 0$ if, and only if, x is a KKT point for (M).

Lemma 24. *Suppose $x \in X$ is not a KKT point for (M), $H = H^T > 0$, and $\eta > 0$. Then* *(i).* $\langle \nabla f_j(x), \hat{d}(x, H, \eta)\rangle < 0$, *for all* $j \in I(x)$, *and*

(ii). $\langle \nabla g_j(x), \hat{d}(x, H, \eta)\rangle < 0$, *for all* $j \in I(x)$.

In order to prove the following lemma we will need to make use of the optimality conditions for $LS^0(x, H, J', I')$, where $x \in X$, $H = H^T > 0$, $J' \subseteq J$, and $I' \subseteq I$. The pair $(\widehat{d^0}, \widehat{\gamma^0})$ is a KKT point for $LS^0(x, H, J', I')$ if there exist multipliers $\widehat{\mu^{0,j}}$, $j \in J'$, and $\widehat{\lambda^{0,j}}$, $j \in I'$, which, together with $(\widehat{d^0}, \widehat{\gamma^0})$, satisfy

$$\begin{cases} \begin{bmatrix} H\widehat{d^0} \\ 1 \end{bmatrix} + \sum_{j \in J'} \widehat{\mu^{0,j}} \begin{bmatrix} \nabla f_j(x) \\ -1 \end{bmatrix} + \sum_{j \in I'} \widehat{\lambda^{0,j}} \begin{bmatrix} \nabla g_j(x) \\ 0 \end{bmatrix} = 0, \\ f_j(x) + \langle \nabla f_j(x), \widehat{d^0}\rangle = F(x) + \widehat{\gamma^0}, \quad \forall j \in J', \\ g_j(x) + \langle \nabla g_j(x), \widehat{d^0}\rangle = 0, \quad \forall j \in I'. \end{cases} \tag{4.4}$$

Lemma 25. *If $\eta_k = 0$, then x_k is a KKT point for (M) and the algorithm will stop in Step 1(ii) at iteration k. On the other hand, whenever the algorithm does not stop in Step 1(ii), the line search is well defined, i.e. Step 2 yields a step $t_k = \beta^j$ for some finite $j = j(k)$.*

Proof. Suppose $\eta_k = 0$. Then $k > 0$ and by *Step 3(iii)* either (i) $\widehat{d_k^0} = 0$, $F(x_k) = F_{\hat{J}_{k-1}}(x_k)$, $\widehat{\mu_k^0} \geq 0$, and $\widehat{\lambda_k^0} \geq 0$, or (ii) $\hat{d}_{k-1} = 0$. Case (ii) cannot hold, otherwise the algorithm would have stopped in *Step 1(ii)* at iteration $k - 1$. Consider case (i). Since $\widehat{d_k^0} = 0$ and $f_j(x_k) = F(x_k)$ for some $j \in \hat{J}_{k-1}$, it follows from the constraints in $LS^0(x_k, H_k, \hat{J}_{k-1}, \hat{I}_{k-1})$ that $\widehat{\gamma_k^0} = 0$. Thus, from the optimality conditions (4.4), it is clear that x_k must be a KKT point for (M) with multipliers

$$\mu^j = \begin{cases} \widehat{\mu_k^{0,j}} & j \in \hat{J}_{k-1}, \\ 0 & \text{otherwise}, \end{cases}$$

and

$$\lambda^j = \begin{cases} \widehat{\lambda_k^{0,j}} & j \in \hat{I}_{k-1}, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, in view of Lemma 23, $\hat{d}_k = 0$ and the algorithm stops in *Step 1(ii)* at iteration k. Thus, if *Step 2* is reached, $\eta_k > 0$ and x_k is not a KKT point. Now, by Lemma 23, $\hat{\gamma}(x_k, H_k, \eta_k) < 0$ and from the first QP constraint $F'(x_k, \hat{d}_k) < 0$. The result follows from Lemma 24 and Assumption 2. □

At this point we have established that the algorithm is well-defined. Further, from Lemma 23, it is clear that if Algorithm **FSQP′-MM** generates a finite sequence terminating at the point x_N, then x_N is a KKT point for the problem (M). Thus, we now focus on the case in which Algorithm **FSQP′-MM** generates an infinite sequence $\{x_k\}$. Note that, in view of Lemma 25, as was the case in Chapter 3, we may assume throughout that

$$\eta_k > 0, \quad \forall k \in \mathbb{N}. \tag{4.5}$$

Lemma 26. *Suppose $\mathcal{K} \subseteq \mathbb{N}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}} x^* \in X$, $H_k \xrightarrow{k\in\mathcal{K}} H^* > 0$, $\{\eta_k\}$ is bounded on $\mathcal{K}$, and $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Then $\hat{J}_k \subseteq J(x^*)$ and $\hat{I}_k \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large and the QP multiplier sequences $\{\hat{\mu}_k\}$ and $\{\hat{\lambda}_k\}$ are bounded on $\mathcal{K}$. Further, given any accumulation point $\eta^* \geq 0$ of $\{\eta_k\}_{k\in\mathcal{K}}$, $(0,0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*)$.*

The analysis that follows will require the Hessian estimate sequence $\{H_k\}$ generated by Algorithm **FSQP′-MM** to be bounded above and bounded away from singularity.

Assumption 4: There exist constants $0 < \sigma_1 \leq \sigma_2$ such that, for all k,

$$\sigma_1 \|d\|^2 \leq \langle d, H_k d \rangle \leq \sigma_2 \|d\|^2, \quad \forall d \in \mathbb{R}^n.$$

Lemma 27. *The sequences* $\{H_k\}$ *and* $\{\eta_k\}$ *generated by Algorithm* **FSQP′-MM** *are bounded. Further, the sequence* $\{\hat{d}_k\}$ *is bounded on subsequences on which* $\{x_k\}$ *is bounded.*

Lemma 28. *If* $\mathcal{K} \subseteq \mathbb{N}$ *is an infinite index set such that* $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$*, then all accumulation points of* $\{x_k\}_{k\in\mathcal{K}}$ *are KKT points for* (M).

We now state and prove the main result of this section. Recall that the proof of Theorem 3 called on the proof of Proposition 3.2 in [48], which was based on a contradiction argument that made heavy use of the line search descent criterion. The merit function for the line search in Algorithm **FSQP′** is the nonsmooth objective function $F(x)$ which is, in a sense, more restrictive than a smooth merit function. It is this fact which complicates the analysis in the following theorem over that in [48].

Theorem 5. *Under the stated assumptions, Algorithm* **FSQP′-MM** *generates a sequence* $\{x_k\}$ *for which all accumulation points are KKT points for* (M).

Proof. Suppose $\mathcal{K} \subseteq \mathbb{N}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}} x^*$. In view of Lemma 27, we may assume without loss of generality that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} \hat{d}^*$, $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \geq 0$, and $H_k \xrightarrow{k\in\mathcal{K}} H^* > 0$. The cases $\eta^* = 0$ and $\eta^* > 0$ are considered separately.

Consider first the case where $\eta^* = 0$. Then, by *Step 3(iii)*, either (i) $\widehat{d_k^0} \xrightarrow{k\in\mathcal{K}} 0$, with $F(x_k) = F_{\hat{J}_{k-1}}(x_k)$, $\widehat{\mu_k^0} \geq 0$, and $\widehat{\lambda_k^0} \geq 0$, for all $k \in \mathcal{K}$, k large enough, or (ii) $\hat{d}_{k-1} \xrightarrow{k\in\mathcal{K}} 0$. Case (ii) implies $x_{k-1} \xrightarrow{k\in\mathcal{K}} x^*$ since $\|x_k - x_{k-1}\| \leq 2\|\hat{d}_{k-1}\| \xrightarrow{k\in\mathcal{K}} 0$. Thus, in view of Lemma 28, x^* is KKT for (M). Now consider case (i). It follows from the optimality conditions (4.4) that $\{\widehat{\mu_k^0}\}_{k\in\mathcal{K}}$ is bounded, thus we assume without loss of generality that $\widehat{\mu_k^0} \xrightarrow{k\in\mathcal{K}} \widehat{\mu^{0,*}}$. We now show that $\hat{I}_{k-1} \subseteq I(x^*)$

for all $k \in \mathcal{K}$, k sufficiently large. Suppose not. Then there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ and an index $\bar{\jmath} \in \hat{I}_{k-1}$, for all $k \in \mathcal{K}'$, such that $\bar{\jmath} \notin I(x^*)$. Thus, $g_{\bar{\jmath}}(x_k) \xrightarrow{k \in \mathcal{K}'} -\delta^* < 0$. In view of our regularity assumptions, and since $\widehat{d_k^0} \xrightarrow{k \in \mathcal{K}'} 0$, this contradicts the constraints in the optimality conditions (4.4) for k sufficiently large. Finally, using the same technique as in Lemma 5 in Chapter 3, we can show that the sequence $\{\widehat{\lambda_k^0}\}_{k \in \mathcal{K}}$ is bounded. Thus, we assume without loss of generality that $\widehat{\lambda_k^0} \xrightarrow{k \in \mathcal{K}} \widehat{\lambda^{0,*}} \geq 0$. Taking limits in the optimality conditions (4.4) on an appropriate subsequence of $\mathcal{K}$ shows that x^* is KKT for (M).

We now turn our attention to the case $\eta^* > 0$. As in the proof of Theorem 3, we argue by contradiction to show that $\hat{d}_k \xrightarrow{k \in \mathcal{K}} 0$. Suppose without loss of generality that there exists $\underline{d} > 0$ such that $\|\hat{d}_k\| \geq \underline{d}$ for all $k \in \mathcal{K}$. Using an argument analogous to that in Theorem 3, we can show that there exists a constant $\delta > 0$ such that for all $k \in \mathcal{K}$, k sufficiently large,

$$F'(x_k, \hat{d}_k) \leq -\delta,$$

$$\langle \nabla g_j(x_k), \hat{d}_k \rangle \leq -\delta, \quad \forall j \in I(x^*)$$

$$g_j(x_k) \leq -\delta, \quad \forall j \in \{1, \ldots, m\} \setminus I(x^*).$$

We now use these inequalities to show that there exists $\underline{t} > 0$ such that $t_k > \underline{t}$ for all $k \in \mathcal{K}$, k sufficiently large. Using the same argument as in [48], we can show that there exists $\underline{t}_j > 0$, $j \in I$, such that

$$g_j(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq 0, \quad \forall t \in [0, \underline{t}_j],\ j \in I.$$

As in [48], for the objectives we will make use of the identity

$$f_j(x_k + t\hat{d}_k + t^2\tilde{d}_k) = f_j(x_k) + \int_0^1 \langle \nabla f_j(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k), t\hat{d}_k + 2t^2\xi\tilde{d}_k \rangle d\xi.$$

Thus,

$$\begin{aligned}
&F(x_k + t\hat{d}_k + t^2\tilde{d}_k) - F(x_k) - \alpha t F'(x_k, \hat{d}_k) \\
&\quad = \max_{j \in J} \left\{ f_j(x_k) + \int_0^1 \langle \nabla f_j(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k), t\hat{d}_k + 2t^2\xi\tilde{d}_k \rangle d\xi \right\} \\
&\qquad -F(x_k) - \alpha t F'(x_k, \hat{d}_k)
\end{aligned}$$

Define $\hat{j} \triangleq \hat{j}(t,k)$ as an index which achieves the first max. Adding and subtracting $tF'(x_k, \hat{d}_k)$, we have

$$\begin{aligned}
&F(x_k + t\hat{d}_k + t^2\tilde{d}_k) - F(x_k) - \alpha t F'(x_k, \hat{d}_k) \\
&\quad = f_{\hat{j}}(x_k) + \int_0^1 \langle \nabla f_{\hat{j}}(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k), t\hat{d}_k + 2t^2\xi\tilde{d}_k \rangle d\xi - F(x_k) \\
&\qquad + tF(x_k) - t \cdot \max_{j \in J} \left\{ f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k \rangle \right\} + (1-\alpha) t F'(x_k, \hat{d}_k) \\
&\quad \leq t \cdot \int_0^1 \langle \nabla f_{\hat{j}}(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k), \hat{d}_k + 2t\xi\tilde{d}_k \rangle - \langle \nabla f_{\hat{j}}(x_k), \hat{d}_k \rangle d\xi \\
&\qquad + (1-\alpha) t F'(x_k, \hat{d}_k) + \underbrace{(1-t)\left(f_{\hat{j}}(x_k) - F(x_k)\right)}_{\leq 0}.
\end{aligned}$$

Thus, for all $k \in \mathcal{K}$, k sufficiently large,

$$\begin{aligned}
&F(x_k + t\hat{d}_k + t^2\tilde{d}_k) - F(x_k) - \alpha t F'(x_k, \hat{d}_k) \\
&\quad \leq t \cdot \left\{ \sup_{\xi \in [0,1]} \|\nabla f_{\hat{j}}(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k) - \nabla f_{\hat{j}}(x_k)\| \cdot \|\hat{d}_k\| \right. \\
&\qquad \left. + 2t \sup_{\xi \in [0,1]} \|\nabla f_{\hat{j}}(x_k + t\xi\hat{d}_k + t^2\xi^2\tilde{d}_k)\| \cdot \|\tilde{d}_k\| - (1-\alpha)\delta \right\}.
\end{aligned}$$

It follows from our regularity assumptions and boundedness of all sequences that there exists $\underline{t}_f > 0$ such that

$$F(x_k + t\hat{d}_k + t^2\tilde{d}_k) - F(x_k) - \alpha t F'(x_k, \hat{d}_k) \leq 0,$$

for all $t \in [0, \underline{t}_f]$, for all $k \in \mathcal{K}$, k sufficiently large. Letting $\underline{t} = \min\{\underline{t}_f, \underline{t}_j, j \in I\}$, we have established that $t_k \geq \underline{t}$ for all $k \in \mathcal{K}$, k sufficiently large. Following the proof of Proposition 3.2 in [48], it is clear from the descent criterion in the line search that

$$\begin{aligned} F(x_{k+1}) &\leq F(x_k) + \alpha t_k F'(x_k, \hat{d}_k) \\ &\leq F(x_k) - \alpha \underline{t}\delta. \end{aligned}$$

On the other hand, since F is continuous, $F(x_k) \overset{k\in\mathcal{K}}{\longrightarrow} F(x^*)$, thus we have a contradiction. This establishes that $\hat{d}_k \overset{k\in\mathcal{K}}{\longrightarrow} 0$. Finally, in view of Lemma 28, we see that x^* is KKT for (M). □

4.3.2 Local Convergence

As usual, in order to establish a result concerning the rate of convergence, we first need to strengthen our regularity assumptions.

Assumption 2′: The functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j \in J$, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j \in I$, are three times continuously differentiable.

Recall that a point x^* is said to satisfy the *second order sufficiency conditions with strict complementary slackness* for (M) if x^* is a regular point (guaranteed by Assumption 3) and if there exist multiplier vectors $\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^m$ such that

- The triple (x^*, μ^*, λ^*) satisfies (4.1), i.e. x^* is a KKT point for (M),
- $\nabla^2_{xx} L(x^*, \mu^*, \lambda^*)$ is positive definite on the subspace

$$\{h \mid \langle \nabla f_i(x^*), h\rangle = \langle \nabla f_j(x^*), h\rangle,\ \forall i, j \in J(x^*)$$
$$\text{and } \langle \nabla g_j(x^*), h\rangle = 0,\ \forall j \in I(x^*)\},$$

- $\mu^{*,j} > 0$ for all $j \in J(x^*)$ and $\lambda^{*,j} > 0$ for all $j \in I(x^*)$ (strict complementary slackness).

Continuing to follow Section 3.3, we strengthen the assumptions one step further in order to ensure that the entire sequence $\{x_k\}$ converges to a KKT point x^*. Again, we have already established, under weaker assumptions, that *every* accumulation point of $\{x_k\}$ is a KKT point for (M).

Assumption 5: The sequence $\{x_k\}$ has an accumulation point x^* which satisfies the second order sufficiency conditions with strict complementary slackness.

A straightforward modification of the proof of Proposition 4.1 in [48] shows that Assumption 5 guarantees the entire sequence converges. We state the result here without proof.

Lemma 29. *The entire sequence generated by Algorithm* **FSQP′-MM** *converges to a point x^* satisfying the second order sufficiency conditions with strict complementary slackness.*

From this point forward, μ^* and λ^* will denote the (unique) multiplier vectors satisfying the KKT conditions for (M) at x^*. Finally, we strengthen our assumptions concerning the sequence $\{H_k\}$.

Assumption 6: The sequence $\{H_k\}$ converges to some $H^* = H^{*T} > 0$.

As was the case in Chapter 3, a major portion of the local convergence analysis will be devoted to showing that $\hat{d}_k$ approaches the SQP direction $d_k^0 =$

$d^0(x_k, H_k)$ sufficiently fast. Note that $QP^0(x_k, H_k)$ is equivalently expressed as

$$\begin{aligned}
\min \quad & \tfrac{1}{2}\langle d^0, H_k d^0\rangle + \gamma^0 \\
\text{s.t.} \quad & f_j(x_k) + \langle \nabla f_j(x_k), d^0\rangle \le F(x_k) + \gamma^0, \quad j \in J, \\
& g_j(x_k) + \langle \nabla g_j(x_k), d^0\rangle \le 0, \qquad\qquad j \in I.
\end{aligned}$$

From this, we readily see that (d_k^0, γ_k^0) is a KKT point for $QP^0(x_k, H_k)$ if there exist multipliers $\mu_k^0 \in \mathbb{R}^p$ and $\lambda_k^0 \in \mathbb{R}^m$ which, together with (d_k^0, γ_k^0), satisfy

$$\left\{\begin{array}{l}
\begin{bmatrix} H_k d_k^0 \\ 1 \end{bmatrix} + \sum_{j\in J} \mu_k^{0,j} \begin{bmatrix} \nabla f_j(x_k) \\ -1 \end{bmatrix} + \sum_{j \in I} \lambda_k^{0,j} \begin{bmatrix} \nabla g_j(x_k) \\ 0 \end{bmatrix} = 0, \\
f_j(x_k) + \langle \nabla f_j(x_k), d_k^0\rangle \le F(x_k) + \gamma_k^0 \cdot \eta, \quad \forall j \in J, \\
g_j(x_k) + \langle \nabla g_j(x_k), d_k^0\rangle \le 0, \quad \forall j \in I, \\
\mu_k^{0,j}\left(f_j(x_k) + \langle \nabla f(x_k), d_k^0\rangle - F(x_k) - \gamma_k^0\right) = 0 \text{ and } \mu_k^{0,j} \ge 0, \quad \forall j \in J \\
\lambda_k^{0,j}\left(g_j(x_k) + \langle \nabla g_j(x_k), d_k^0\rangle\right) = 0 \text{ and } \lambda^{0,j} \ge 0, \quad \forall j \in I.
\end{array}\right. \tag{4.6}$$

Define the active sets for $QP^0(x_k, H_k)$

$$\begin{aligned}
J_k^0 &\triangleq \{\, j \in J \mid f_j(x_k) + \langle \nabla f_j(x_k), d_k^0\rangle = F(x_k) + \gamma_k^0 \,\}, \\
I_k^0 &\triangleq \{\, j \in I \mid g_j(x_k) + \langle \nabla g_j(x_k), d_k^0\rangle = 0 \,\}.
\end{aligned}$$

The following lemma is proved similarly to Lemma 3 in [47].

Lemma 30.

(i) $d_k^0 \to 0$*,*

(ii) $\mu_k^0 \to \mu^*$ *and* $\lambda_k^0 \to \lambda^*$*.*

(iii) For all k sufficiently large, the following equalities hold

$$J_k^0 = \{ \ j \in J \mid \mu_k^{0,j} > 0 \ \} = J(x^*)$$
$$I_k^0 = \{ \ j \in I \mid \lambda_k^{0,j} > 0 \ \} = I(x^*).$$

In the analysis that follows, we will be assuming that $|J(x^*)| > 1$, i.e. more than one objective is active at the solution. The motivation for this assumption is primarily ease of notation as well as the fact that it allows us to ignore several special cases. When $|J(x^*)| = 1$, since we have shown $x_k \to x^*$, it is easily checked that the analysis from Chapter 3 allows us to conclude 2-step superlinear convergence. Furthermore, without loss of generality, we will be assuming that

$$J(x^*) = \{ \ 1, \ldots, r \ \},$$

where $1 < r \leq p$. Now define

$$R_k \triangleq [\ \nabla f_2(x_k) - \nabla f_1(x_k), \ldots, \nabla f_r(x_k) - \nabla f_1(x_k), \nabla g_j(x_k) \ : \ j \in I(x^*) \ ],$$
$$g_k \triangleq [\ f_2(x_k) - f_1(x_k), \ldots, f_r(x_k) - f_1(x_k), g_j(x_k) \ : \ j \in I(x^*) \ ]^T.$$

The following assumption was not needed in Chapter 3, as it followed immediately from previous assumptions, but it is crucial for the analysis that follows.

Assumption 7: The matrix

$$R^* \triangleq \lim_{k \to \infty} R_k$$

has full rank.

Note that since $x_k \to x^*$, R^* is well-defined. This assumption allows us to generalize Lemma 10 to the present case. As it is a standard result, we state it without proof.

Lemma 31. *Under the stated assumptions, the matrix*

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix}$$

is uniformly invertible, i.e. it has bounded condition number for all k.

The importance of the matrix given in Lemma 31 comes from its relationship to the SQP direction $d_k^0 = d^0(x_k, H_k)$ for all k sufficiently large.

Lemma 32. *For all k sufficiently large, there exists $0 < \psi_k^0 \in \mathbb{R}^{r-1+|I(x^*)|}$ such that, together with $d_k^0 = d^0(x_k, H_k)$, $\begin{pmatrix} d_k^0 \\ \psi_k^0 \end{pmatrix}$ is the unique solution of the linear system*

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d \\ \psi \end{pmatrix} = - \begin{pmatrix} \nabla f_1(x_k) \\ g_k \end{pmatrix}.$$

Proof. In view of the optimality conditions (4.6) and Lemma 30(*iii*), for all k sufficiently large we have

$$\begin{aligned} f_1(x_k) + \langle \nabla f_1(x_k), d_k^0 \rangle &= \gamma_k^0 + F(x_k) \\ &= f_j(x_k) + \langle \nabla f_j(x_k), d_k^0 \rangle, \quad j = 2, \dots, r. \end{aligned}$$

Thus, for all k sufficiently large,

$$\begin{aligned} \langle \nabla f_j(x_k) - \nabla f_1(x_k), d_k^0 \rangle &= - \left(f_j(x_k) - f_1(x_k) \right), \quad j = 2, \dots, r, \\ \langle \nabla g_j(x_k), d_k^0 \rangle &= -g_j(x_k), \quad j \in I(x^*). \end{aligned} \tag{4.7}$$

Adding and subtracting

$$\nabla f_1(x_k) = \left(\sum_{j=1}^{r} \mu_k^{0,j} \right) \cdot \nabla f_1(x_k)$$

from (the first n elements of) the first equation of the optimality conditions (4.6), and using complementary slackness, gives

$$H_k d_k^0 + \sum_{j=2}^{r} \mu_k^{0,j} \langle \nabla f_j(x_k) - \nabla f_1(x_k), d_k^0 \rangle + \sum_{j \in I(x^*)} \lambda_k^{0,j} \langle \nabla g_j(x_k), d_k^0 \rangle = -\nabla f_1(x_k). \tag{4.8}$$

Combining (4.8) with (4.7) and defining

$$\psi_k^0 \triangleq [\ \mu_k^{0,j},\ j = 2, \ldots, r,\ \ \lambda_k^{0,j},\ j \in I(x^*)]^T$$

gives the result. □

The modification of Powell's superlinear convergence result used in Section 3.3.2 does not directly apply to mini-max problems. On the other hand, note that under the current assumptions x^* also satisfies the strong second order sufficiency conditions with strict complementary slackness for the smooth problem

$$\begin{array}{lll} \min & f_1(x) & \\ \text{s.t.} & f_j(x) - f_1(x) \leq 0, & j = 2, \ldots, r, \\ & g_j(x) \leq 0, & j \in I(x^*), \end{array} \tag{E}$$

where all constraints are active at the solution x^*. It is easily verified that (E) satisfies all of the necessary assumptions given in Section 2 of [60]. Now suppose that $d^e(x_k, H_k)$ is the SQP direction as computed for (E) on the sequences $\{x_k\}$ and $\{H_k\}$ generated by **FSQP′-MM** for (M). Using Lemma 32, it is straightforward to show that, for all k sufficiently large,

$$d^e(x_k, H_k) = d^0(x_k, H_k).$$

Thus, it should be clear that we may equivalently assume our algorithm is iterating on the smooth problem (E). It remains to establish that, for the iteration on (E),

1. The multiplier sequences converge to the true multipliers,
2. For all k sufficiently large, the binding sets for the QPs correspond to the active sets at the solution x^*,
3. $x_{k+1} - x_k = O(\|d_k^0\|^2)$, and
4. $t_k = 1$ for all k sufficiently large.

Then, our modification of Powell's argument as discussed in Chapter 3 may be applied to establish 2-step superlinear convergence for the mini-max algorithm. While the proofs are occasionally different, the following sequence of lemmas are direct extensions of those in Section 3.3.2.

Given $\eta^* \geq 0$, we extend the definition of the set $N^*(\eta^*)$ given in Section 3.3.2 to

$$N^*(\eta^*) \triangleq \left\{ \begin{pmatrix} \nabla f_j(x^*) \\ -1 \end{pmatrix}, j = 1, \dots, r; \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix}, j \in I(x^*) \right\}.$$

Lemma 33. *Given any $\eta^* \geq 0$, the set $N^*(\eta^*)$ is linearly independent.*

Proof. Note that, in view of Lemma 23, $\hat{d}^* = \hat{d}(x^*, H^*, \eta^*) = 0$. Now suppose the claim does not hold, i.e. suppose there exist scalars μ^j, $j = 1, \dots, r$, and λ^j, $j \in I(x^*)$, not all zero, such that

$$\sum_{j=1}^{r} \mu^j \begin{pmatrix} \nabla f_j(x^*) \\ -1 \end{pmatrix} + \sum_{j \in I(x^*)} \lambda^j \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix} = 0. \tag{4.9}$$

In view of Assumption 3, μ^j, $j = 1, \dots, r$, are not all zero and λ^j, $j \in I(x^*)$, are not all zero. First consider the case $\sum_{j=1}^{r} \mu^j \neq 0$, and assume without loss of generality (may need to divide (4.9) by $\sum_{j=1}^{r} \mu^j$) that

$$\sum_{j=1}^{r} \mu^j = 1.$$

Then, it follows from (4.9) that

$$\nabla f_1(x^*) + \sum_{j=2}^{r} \mu^j \left(\nabla f_j(x^*) - \nabla f_1(x^*)\right) + \sum_{j \in I(x^*)} \lambda^j \nabla g_j(x^*) = 0,$$

and

$$\eta^* \cdot \sum_{j \in I(x^*)} \lambda^j = -1. \tag{4.10}$$

In view of Assumption 7, μ^j, $j = 1, \ldots, r$, and λ^j, $j \in I(x^*)$, are the unique multipliers for (M), and, by complementary slackness, are all positive. As this contradicts (4.10), we must have $\sum_{j=1}^{r} \mu^j = 0$. In this case, (4.9) gives

$$\sum_{j=2}^{r} \mu^j \left(\nabla f_j(x^*) - \nabla f_1(x^*)\right) + \sum_{j \in I(x^*)} \lambda^j \nabla g_j(x^*) = 0,$$

which immediately contradicts Assumption 7. Thus, $N^*(\eta^*)$ is linearly independent. □

Lemma 34. *Let $\eta^* \geq 0$ be an accumulation point of $\{\eta_k\}$. Then $(\hat{d}^*, \hat{\gamma}^*) = (0, 0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*)$ and the second order sufficiency conditions hold, with strict complementary slackness.*

Proof. In view of Lemma 23, $\widehat{QP}(x^*, H^*, \eta^*)$ has $(\hat{d}^*, \hat{\gamma}^*) = (0, 0)$ as its unique solution. A straightforward modification of the proof of Lemma 12 shows that the second-order sufficiency conditions hold. It remains to show that strict complementary slackness holds. Let $\hat{J}^* \subseteq J$ and $\hat{I}^* \subseteq I$ denote the active sets at the solution of $\widehat{QP}(x^*, H^*, \eta^*)$. Since $(\hat{d}^*, \hat{\gamma}^*) = (0, 0)$, it should be clear that $\hat{J}^* = \{1, \ldots, r\}$ and $\hat{I}^* = I(x^*)$. Suppose that $\hat{\mu}^{*,j}$, $j = 1, \ldots, r$, and $\hat{\lambda}^{*,j}$, $j \in I(x^*)$ are the multipliers satisfying (4.2) at $(0, 0)$. Suppose that $\sum_{j=1}^{r} \hat{\mu}^{*,j} = 0$. Then it follows from (4.2) that

$$\sum_{j=2}^{r} \hat{\mu}^{*,j} \left(\nabla f_j(x^*) - \nabla f_1(x^*)\right) + \sum_{j \in I(x^*)} \hat{\lambda}^{*,j} \nabla g_j(x^*) = 0.$$

As this contradicts Assumption 7, we must have $\bar{\mu} \triangleq \sum_{j=1}^{r} \hat{\mu}^{*,j} > 0$. Now, it is clear from (4.2) and uniqueness of the multipliers $\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^m$ that

$$\mu^{*,j} = \begin{cases} \hat{\mu}^{*,j}, & j = 1, \ldots, r, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\lambda^{*,j} = \begin{cases} \hat{\lambda}^{*,j}, & j \in I(x^*), \\ 0, & \text{otherwise.} \end{cases}$$

Thus, by Assumption 5 (strict complementary slackness for (M)), it follows that strict complementary slackness holds for $\widehat{QP}(x^*, H^*, \eta^*)$. □

Just as for Lemma 13 in Chapter 3, the following lemma now follows from Theorem 2.1 in [68].

Lemma 35. *If $\mathcal{K}$ is a subsequence on which $\{\eta_k\}$ converges, say to $\eta^* \geq 0$, then $\hat{\mu}_k \overset{k\in\mathcal{K}}{\longrightarrow} \left(\sum_{j=1}^{p} \hat{\mu}^{*,j}\right) \mu^* > 0$ and $\hat{\lambda}_k \overset{k\in\mathcal{K}}{\longrightarrow} \hat{\mu}^* \cdot \lambda^*$, where $\hat{\mu}^* = \hat{\mu}^*(\eta^*)$ is the KKT multiplier vector for the constraints corresponding to the true objectives in $\widehat{QP}(x^*, H^*, \eta^*)$. Finally, $\hat{d}_k \to 0$ and $\hat{\gamma}_k \to 0$.*

A trivial extension of Lemma 14 shows that $\widehat{QP}(x_k, H_k, \eta_k)$ eventually correctly identifies the active set at the solution of (M).

Lemma 36. *For all k sufficiently large, $\hat{J}_k = \{1, \ldots, r\}$ and $\hat{I}_k = I(x^*)$.*

We can now show that our estimate of the SQP direction is exact for all k large enough.

Lemma 37. *For all k sufficiently large, $\widehat{d_k^0} = d_k^0$.*

Proof. In view of Lemma 36 and the optimality conditions 4.4, for all k sufficiently large

$$\begin{gathered}H_k\widehat{d_k^0} + \sum_{j=1}^{r} \widehat{\mu_k^{0,j}} \nabla f_j(x_k) + \sum_{j \in I(x^*)} \widehat{\lambda_k^{0,j}} \nabla g_j(x_k) = 0, \\ \sum_{j=1}^{r} \widehat{\mu_k^{0,j}} = 1,\end{gathered} \tag{4.11}$$

and

$$\widehat{\gamma_k^0} = f_j(x_k) - F(x_k) + \langle \nabla f_j(x_k), \widehat{d_k^0} \rangle, \quad j = 1, \dots, r.$$

It follows that

$$f_j(x_k) - f_1(x_k) + \langle \nabla f_j(x_k) - \nabla f_1(x_k), \widehat{d_k^0} \rangle = 0, \quad j = 2, \dots, r.$$

Subtracting $\nabla f_1(x_k) = \left(\sum_{j=1}^{r} \widehat{\mu_k^{0,j}} \right) \nabla f_1(x_k)$ from the first equation in (4.11) and defining

$$\widehat{\psi_k^0} \triangleq \left(\widehat{\mu_k^{0,j}},\ j = 2, \dots, r;\ \widehat{\lambda_k^{0,j}},\ j \in I(x^*) \right)^T,$$

we have

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} \widehat{d_k^0} \\ \widehat{\psi_k^0} \end{pmatrix} = - \begin{pmatrix} \nabla f_1(x_k) \\ g_k \end{pmatrix},$$

for all k sufficiently large. The result then follows from Lemmas 32 and 31. □

Lemma 38.

(i) $\eta_k \to 0$,

(ii) $\hat{\mu}_k \to 1$, *and* $\hat{\lambda}_k \to \lambda^*$.

(iii) For all k *sufficiently large,* $\hat{I}_k = \{ \ j \mid \hat{\lambda}_k^j > 0 \ \}$.

Proof. Straightforward extension of proof of Lemma 16. □

Lemma 39.

(i) $\eta_k = O(\|d_k^0\|^2)$,

(ii) $\hat{d}_k = d_k^0 + O(\|d_k^0\|^2)$.

(iii) $\hat{\gamma}_k = O(\|d_k^0\|)$.

Proof. The proofs of (i) and (ii) are straightforward extensions of the proofs of Lemma 17(i) and (ii). In view of Lemma 36, for all k sufficiently large,

$$\hat{\gamma}_k = \langle \nabla f_j(x_k), \hat{d}_k \rangle + f_j(x_k) - F(x_k), \quad j = 1, \dots, r.$$

Further, since $x_k \to x^*$, $J(x_k) \subseteq J(x^*)$ for all k sufficiently large. Thus, for all k sufficiently large we may choose $\hat{j}(k) \in J(x_k)$ such that $\hat{j}(k) \in J(x^*)$. Clearly, $f_{\hat{j}(k)} - F(x_k) = 0$ for all k sufficiently large, thus

$$\hat{\gamma}_k = \langle \nabla f_{\hat{j}(k)}(x_k), \hat{d}_k \rangle,$$

for all k sufficiently large. It follows that

$$\hat{\gamma}_k = O(\|\hat{d}_k\|) = O(\|d_k^0\|).$$

□

Lemma 40. $\tilde{d}_k = O(\|d_k^0\|^2)$.

Proof. In view of Lemma 36 and the constraints in $\widetilde{LS}(x_k, \hat{d}_k, H_k, J_k, I_k)$, we have

$$f_j(x_k + \hat{d}_k) - f_1(x_k + \hat{d}_k) + \langle \nabla f_j(x_k) - \nabla f_1(x_k), \tilde{d}_k \rangle = 0, \quad j = 2, \dots, r.$$

Further, from the constraints of $\widehat{QP}(x_k, H_k, \eta_k)$,

$$f_j(x_k) - f_1(x_k) + \langle \nabla f_j(x_k) - \nabla f_1(x_k), \hat{d}_k \rangle = 0, \quad j = 2, \dots, r.$$

for all k sufficiently large. Expanding $f_j - f_1$ about x_k and evaluating at $x_k + \hat{d}_k$ gives

$$\overbrace{f_j(x_k) - f_1(x_k) + \langle \nabla f_j(x_k) - \nabla f_1(x_k), \hat{d}_k\rangle}^{=\ 0} + \langle \nabla f_j(x_k) - \nabla f_1(x_k), \tilde{d}_k\rangle$$
$$+\frac{1}{2}\langle \hat{d}_k, (\nabla^2 f_j(x_k + \xi_k^j \hat{d}_k) - \nabla^2 f_1(x_k + \xi_k^j \hat{d}_k))\hat{d}_k\rangle = 0,$$

for some $\xi_k^j \in (0,1)$. Thus,

$$\begin{aligned}\langle \nabla f_j(x_k) - \nabla f_1(x_k), \tilde{d}_k\rangle &= -\frac{1}{2}\langle \hat{d}_k, (\nabla^2 f_j(x_k + \xi_k^j \hat{d}_k) - \nabla^2 f_1(x_k + \xi_k^j \hat{d}_k))\hat{d}_k\rangle \\ &= O(\|\hat{d}_k\|^2) = O(\|d_k^0\|^2).\end{aligned}$$

Using the same argument as in Lemma 18, we can similarly show

$$\langle \nabla g_j(x_k), \tilde{d}_k\rangle = O(\|d_k^0\|^2), \quad j \in I(x^*).$$

Combining results, we have established

$$R_k^T \tilde{d}_k = O(\|d_k^0\|^2).$$

The rest of the proof is a straightforward extension of that for Lemma 18. □

As in Section 3.3.2, we now add one additional assumption to ensure that the matrices $\{H_k\}$ suitably approximate the Hessian of the Lagrangian at the solution. Define the projection

$$P_k \triangleq I - R_k(R_k^T R_k)^{-1} R_k^T.$$

Assumption 8:

$$\lim_{k\to\infty} \frac{\|P_k(H_k - \nabla_{xx}^2 L(x^*, \lambda^*))P_k \hat{d}_k\|}{\|\hat{d}_k\|} = 0.$$

Lemma 41. *There exist constants $\nu_1,\ \nu_2,\ \nu_3 > 0$ such that* *(i)* $F'(x_k, \hat{d}_k) \leq -\nu_1 \|d_k^0\|^2$,

(ii) for all k sufficiently large

$$\sum_{j=1}^{m} \hat{\lambda}_k^j g_j(x_k) \leq -\nu_2 \|g_k\|,$$

(iii) $\hat{d}_k = P_k \hat{d}_k + d_k^1$, *where*

$$\|d_k^1\| \leq \nu_3 \|g_k\| + O(\|d_k^0\|^3),$$

for all k sufficiently large.

Proof. From the constraints of $\widehat{QP}(x_k, H_k, \eta_k)$, we have

$$\begin{array}{rcl} f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k \rangle - F(x_k) & \leq & \hat{\gamma}_k \\ & \leq & -\dfrac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle \\ & \leq & -\dfrac{\sigma_1}{2}\|\hat{d}_k\|^2. \end{array}$$

Thus, for all k we have

$$\begin{array}{rcl} F'(x_k, \hat{d}_k) & = & \max\limits_j \{f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k \rangle - F(x_k)\} \\ & \leq & -\dfrac{\sigma_1}{2}\|\hat{d}_k\|^2, \end{array}$$

and (i) follows. Claim (ii) may be proved using a straightforward extension of the proof of Lemma 19(ii). In view of the optimality conditions 4.2 and Lemma 36,

$$R_k^T \hat{d}_k = -g_k - \hat{\gamma}_k \eta_k \begin{pmatrix} \mathbf{0}_{r-1} \\ \mathbf{1}_{|I(x^*)|} \end{pmatrix},$$

where $\mathbf{0}_{r-1}$ is the vector of $r-1$ zeros and $\mathbf{1}_{|I(x^*)|}$ is the vector of $|I(x^*)|$ ones. Using this equation and Assumption 7, we may apply the same argument as was used for Lemma 19(iii) to show that Claim (iii) holds. □

Lemma 42. *For all k sufficiently large, $t_k = 1$.*

Proof. The same argument as was used in the proof of Lemma 20 may be used to show that the constraints are satisfied with a step of one for all k sufficiently large. In order to show that the descent criterion is satisfied we loosely follow the proof given in [47]. For $j \in \{1, \ldots, r\}$, expanding f_j about $x_k + \hat{d}_k$ we have

$$f_j(x_k+\hat{d}_k+\tilde{d}_k) = f_j(x_k+\hat{d}_k)+\langle \nabla f_j(x_k+\hat{d}_k), \tilde{d}_k\rangle + \frac{1}{2}\langle \tilde{d}_k, \nabla^2 f_j(x_k+\hat{d}_k+\xi_k^j\tilde{d}_k)\tilde{d}_k\rangle,$$

for some $\xi_k^j \in (0, 1)$. Now expanding ∇f_j about x_k and using Lemmas 39 and 40,

$$f_j(x_k + \hat{d}_k + \tilde{d}_k) = \underbrace{f_j(x_k + \hat{d}_k) + \langle \nabla f_j(x_k), \tilde{d}_k\rangle}_{=\tilde{\gamma}_k+F_{\{1,\ldots,r\}}(x_k+\hat{d}_k)} + O(\|d_k^0\|^3), \quad j = 1, \ldots, r. \tag{4.12}$$

Thus,

$$f_j(x_k + \hat{d}_k + \tilde{d}_k) = f_i(x_k + \hat{d}_k + \tilde{d}_k) + O(\|d_k^0\|^3), \quad i, j = 1, \ldots, r.$$

Note that, since $x_k + \hat{d}_k + \tilde{d}_k \to x^*$,

$$F(x_k + \hat{d}_k + \tilde{d}_k) = F_{\{1,\ldots,r\}}(x_k + \hat{d}_k + \tilde{d}_k), \tag{4.13}$$

for all k sufficiently large. Define $\bar{\mu}_k \overset{\Delta}{=} \sum_{j=1}^r \hat{\mu}_k^j$. Then, in view of Lemma 38(*ii*) and strict complementary slackness, $\bar{\mu}_k > 0$ for all k sufficiently large. Using (4.12), (4.13), and the fact that $\sum_{j=1}^r (\hat{\mu}_k^j/\bar{\mu}_k) = 1$,

$$F(x_k + \hat{d}_k + \tilde{d}_k) = \sum_{j=1}^r \frac{\hat{\mu}_k^j}{\bar{\mu}_k} f_j(x_k + \hat{d}_k + \tilde{d}_k) + O(\|d_k^0\|^3).$$

Expanding f_j about x_k in the above expression we get (after a little algebra)

$$F(x_k + \hat{d}_k + \tilde{d}_k) = \sum_{j=1}^r \frac{\hat{\mu}_k^j}{\bar{\mu}_k}(\overbrace{f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k\rangle}^{=\hat{\gamma}_k+F(x_k)}$$
$$+\langle \nabla f_j(x_k), \tilde{d}_k\rangle + \frac{1}{2}\langle \hat{d}_k, \nabla^2 f_j(x_k)\hat{d}_k\rangle) + O(\|d_k^0\|^3).$$

Using the fact that $\hat{\gamma}_k = F'(x_k, \hat{d}_k)$ and rearranging, we have

$$\begin{aligned} F(x_k + \hat{d}_k + \tilde{d}_k) &= F(x_k) + \frac{1}{2}F'(x_k, \hat{d}_k) \\ &+ \frac{1}{2}\sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k}\Big(f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k\rangle - F(x_k)\Big) \\ &+ \sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k}\left(\langle \nabla f_j(x_k), \tilde{d}_k\rangle + \frac{1}{2}\langle \hat{d}_k, \nabla^2 f_j(x_k)\hat{d}_k\rangle\right) \\ &+ O(\|d_k^0\|^3). \end{aligned}$$

As $f_j(x_k) - F(x_k) \leq 0$, the above expression gives

$$\begin{aligned} F(x_k + \hat{d}_k + \tilde{d}_k) - F(x_k) &\leq \frac{1}{2}F'(x_k, \hat{d}_k) + \frac{1}{2}\sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k}\langle \nabla f_j(x_k), \hat{d}_k\rangle \\ &+ \sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k}\left(\langle \nabla f_j(x_k), \tilde{d}_k\rangle + \frac{1}{2}\langle \hat{d}_k, \nabla^2 f_j(x_k)\hat{d}_k\rangle\right) \\ &+ O(\|d_k^0\|^3). \end{aligned} \tag{4.14}$$

From the optimality conditions (4.2), for all k sufficiently large

$$H_k\hat{d}_k + \sum_{j=1}^{r} \hat{\mu}_k^j \nabla f_j(x_k) + \sum_{j\in I(x^*)} \hat{\lambda}_k^j \nabla g_j(x_k) = 0.$$

Taking the inner product of the above equation with $\hat{d}_k$ and $\tilde{d}_k$ respectively gives

$$\sum_{j=1}^{r} \hat{\mu}_k^j \langle \nabla f_j(x_k), \hat{d}_k\rangle = -\sum_{j\in I(x^*)} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \hat{d}_k\rangle - \langle \hat{d}_k, H_k\hat{d}_k\rangle, \tag{4.15}$$

and

$$\sum_{j=1}^{r} \hat{\mu}_k^j \langle \nabla f_j(x_k), \tilde{d}_k\rangle = -\sum_{j\in I(x^*)} \hat{\lambda}_k^j \langle \nabla g_j(x_k), \tilde{d}_k\rangle + O(\|d_k^0\|^3). \tag{4.16}$$

Plugging (4.15) and (4.16) into (4.14), after dividing by $\bar{\mu}_k$, we find

$$\begin{aligned} F(x_k + \hat{d}_k + \tilde{d}_k) - F(x_k) &\leq \frac{1}{2}F'(x_k, \hat{d}_k) - \frac{1}{2}\sum_{j\in I(x^*)} \frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle \nabla g_j(x_k), \hat{d}_k\rangle \\ &- \sum_{j\in I(x^*)} \frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle \nabla g_j(x_k), \tilde{d}_k\rangle - \frac{1}{2\bar{\mu}_k}\langle \hat{d}_k, H_k\hat{d}_k\rangle \\ &+ \frac{1}{2}\sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k}\langle \hat{d}_k, \nabla^2 f_j(x_k)\hat{d}_k\rangle + O(\|d_k^0\|^3). \end{aligned}$$

Add and subtract

$$\frac{1}{2}\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\nabla g_j(x_k),\hat{d}_k\rangle=\frac{1}{2}\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}(\underbrace{\hat{\gamma}_k\cdot\eta_k}_{=O(\|d_k^0\|^3)}-g_j(x_k))$$

from the right hand side of the previous expression, yielding

$$\begin{aligned}F(x_k+\hat{d}_k+\tilde{d}_k)-F(x_k)\quad\leq\quad&\frac{1}{2}F'(x_k,\hat{d}_k)-\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\nabla g_j(x_k),\hat{d}_k\rangle\\&-\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\nabla g_j(x_k),\tilde{d}_k\rangle-\frac{1}{2}\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}g_j(x_k)\\&-\frac{1}{2\bar{\mu}_k}\langle\hat{d}_k,H_k\hat{d}_k\rangle+\frac{1}{2}\sum_{j=1}^{r}\frac{\hat{\mu}_k^j}{\bar{\mu}_k}\langle\hat{d}_k,\nabla^2 f_j(x_k)\hat{d}_k\rangle\\&+O(\|d_k^0\|^3).\end{aligned}\tag{4.17}$$

Using the exactly the same argument as in the proof of Lemma 20, we can show

$$g_j(x_k+\hat{d}_k+\tilde{d}_k)=O(\|d_k^0\|^\tau),\quad j\in I(x^*).$$

Thus, expanding about x_k, we have

$$g_j(x_k)+\langle\nabla g_j(x_k),\hat{d}_k\rangle+\langle\nabla g_j(x_k),\tilde{d}_k\rangle+\frac{1}{2}\langle\hat{d}_k,\nabla^2 g_j(x_k)\hat{d}_k\rangle=O(\|d_k^0\|^\tau),$$

for $j\in I(x^*)$. Multiplying this expression by $\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}$ and summing over $I(x^*)$ gives

$$\begin{aligned}&-\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\nabla g_j(x_k),\hat{d}_k\rangle-\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\nabla g_j(x_k),\tilde{d}_k\rangle\\&\qquad=\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}g_j(x_k)+\frac{1}{2}\sum_{j\in I(x^*)}\frac{\hat{\lambda}_k^j}{\bar{\mu}_k}\langle\hat{d}_k,\nabla^2 g_j(x_k)\hat{d}_k\rangle+O(\|d_k^0\|^\tau).\end{aligned}$$

So, substituting this and then subtracting $\alpha F'(x_k,\hat{d}_k)$ from both sides of (4.17) we see

$$
\begin{aligned}
F(x_k + \hat{d}_k + \tilde{d}_k) - F(x_k) - \alpha F'(x_k, \hat{d}_k) &\le \left(\frac{1}{2} - \alpha\right) F'(x_k, \hat{d}_k) \\
&+ \frac{1}{2}\left\langle \hat{d}_k, \left(\sum_{j=1}^{r} \frac{\hat{\mu}_k^j}{\bar{\mu}_k} \nabla^2 f_j(x_k) + \sum_{j \in I(x^*)} \frac{\hat{\lambda}_k^j}{\bar{\mu}_k} \nabla^2 g_j(x_k) - H_k\right) \hat{d}_k \right\rangle \\
&+ \frac{1}{2} \sum_{j \in I(x^*)} \frac{\hat{\lambda}_k^j}{\bar{\mu}_k} g_j(x_k) + O(\|d_k^0\|^\tau).
\end{aligned}
$$

As $\bar{\mu}_k$ is bounded away from zero for all k sufficiently large, the rest of the proof follows that of Lemma 20. □

Theorem 6. *Algorithm* **FSQP′** *generates a sequence* $\{x_k\}$ *which converges 2-step superlinearly to* x^**, i.e.*

$$
\lim_{k \to \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.
$$

4.4 Implementation and Numerical Results

The implementation details for Algorithm **FSQP′-MM** are exactly the same, or direct extensions of, those for Algorithm **FSQP′** as given in Section 3.4. The active objectives from $\widehat{QP}(x_k, H_k, \eta_k)$ will be taken as those in the set

$$
\hat{J}_k = \{\ j \in J \mid f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k \rangle - F(x_k) - \hat{\gamma}_k > -\sqrt{\epsilon_m}\ \},
$$

where ϵ_m is the machine precision. The tilting parameter scaling factors C_k^j, $j \in I$, are updated following the same rule, but now C_k^j is decreased when constraint g_j did not cause a line search failure and *some* objective did, i.e.

```
**if** ($g_j(\cdot)$ caused line search failure) **then** $C^j_{k+1} \leftarrow C^j_k \cdot \delta_c$
**else if** (some $f_i(\cdot)$ caused line search failure) **then** $C^j_{k+1} \leftarrow C^j_k / \delta_c$
**if** ($C^j_{k+1} < \underline{C}$) **then** $C^j_{k+1} \leftarrow \underline{C}$
**if** ($C^j_{k+1} > \overline{C}$) **then** $C^j_{k+1} \leftarrow \overline{C}$
```

Finally, for the Hessian update, in order to provide a better estimate of the multipliers at the solution, if $\sum_{j\in J} \hat{\mu}^j_k > \sqrt{\epsilon_m}$, we again normalize the multipliers as follows

$$\hat{\mu}^j_k \leftarrow \frac{\hat{\mu}^j_k}{\sum_{j\in J} \hat{\mu}^j_k}, \quad j \in J,$$

$$\hat{\lambda}^j_k \leftarrow \frac{\hat{\lambda}^j_k}{\sum_{j\in J} \hat{\mu}^j_k}, \quad j \in I.$$

Note that the quantity γ_{k+1} is now defined as

$$\gamma_{k+1} \stackrel{\Delta}{=} \nabla_x L(x_{k+1}, \hat{\mu}_k, \hat{\lambda}_k) - \nabla_x L(x_k, \hat{\mu}_k, \hat{\lambda}_k).$$

Otherwise, the Hessian update is precisely as given in Section 3.4.

In order to test Algorithm **FSQP′-MM** we chose several problems from the literature. Unable to find good nonlinearly constrained mini-max test problems, following [36, 79] we took problems `43`, `84`, `113`, and `117` from [28] and turned them into mini-max problems by removing some constraints and adding objectives of the form

$$f_i(x) = f(x) + \alpha_i g_i(x),$$

where $\alpha_i > 0$ are fixed scalars. Specifically, for `p43m`, the first two constraints of problem `43` were removed and converted into objectives using a value of $\alpha_i = 15$ for both. For `p84m`, constraints 5 and 6 of problem `84` were removed and converted into objectives using $\alpha_i = 20$ for both. Next, for `p113m` the first three linear constraints of problem 113 were converted into objectives using $\alpha_i = 10$ for all. Finally, for p117m, the first two nonlinear constraints of problem p117 were converted into objective functions with $\alpha_i = 10$ for both. Problem dav is from [75], problems polk1 - polk4 are from [56], and problem kiwi1 is from [33]. Finite difference gradients were used for all test problems except polk1 - polk4, where analytic gradients were used.

In Table 4.1, we give the results for **FSQP′-MM**, which we call NEW in the table, and CFSQP [36], which implements a mini-max extension (similar to that discussed in this chapter) of the algorithm **FSQP** [51]. The first column gives the specific problem being solved and the column labeled ALGO tells which algorithm was used to solve the given problem. The next three columns indicate the size of the problem following the notation of this chapter (for all problems, m indicates the number of *nonlinear* constraints). The columns labeled NF, NG, and IT give the number of (scalar) objective function evaluations, nonlinear constraint function evaluations, and iterations required to solve the problem, respectively. Finally, $F(x^*)$ is the value of the maximum objective function at the final iterate and ϵ is the tolerance for the size of the search direction (the stopping criterion).

Again the numerical results are very encouraging. On average, our implementation of Algorithm **FSQP′-MM** seems to take the same number of iterations and function evaluations as CFSQP. Given that the cost to generate a new iterate is much cheaper for Algorithm **FSQP′-MM**, the results seem to indicate that the new algorithm may be superior for applications in which the cost of evaluating functions is dominated by the cost of generating a new iterate, and it is at least as good for other applications.

	ALGO	n	p	m	NF	NG	IT	$F(x^*)$	ϵ
p43m	NEW	4	3	1	48	23	10	-4.40000e+1	5e-6
	CFSQP				67	32	13	-4.40000e+1	
p84m	NEW	5	3	4	58	48	12	-5.28034e+6	5e-6
	CFSQP				17	20	4	-5.28034e+6	
p113m	NEW	10	4	5	109	125	14	2.43062e+1	5e-6
	CFSQP				108	127	14	2.43062e+1	
p117m	NEW	15	3	3	97	103	17	3.23487e+1	5e-6
	CFSQP				124	144	21	3.23487e+1	
dav	NEW	4	20	0	272		12	1.15706e+2	5e-6
	CFSQP				342		12	1.15706e+2	
polk1	NEW	2	2	0	91		23	2.71830e0	5e-6
	CFSQP				42		11	2.71828e0	
polk2	NEW	10	2	0	184		34	5.45982e+1	5e-6
	CFSQP				217		45	5.45982e+1	
polk3	NEW	11	10	0	191		15	3.70348e0	5e-6
	CFSQP				236		17	3.70348e0	
polk4	NEW	2	3	0	46		8	1.36429e-5	5e-6
	CFSQP				45		8	4.09939e-7	
kiwi1	NEW	5	10	0	180		13	2.26002e+1	1e-6
	CFSQP				159		11	2.26002e+1	

Table 4.1: Numerical results for **FSQP′-MM**.

Chapter 5

Discretized Problems from Semi-Infinite Programming

5.1 Introduction

Consider the Semi-Infinite Programming (SIP) problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \Phi(x) \leq 0, \end{aligned} \tag{SI}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, and $\Phi : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\Phi(x) \stackrel{\Delta}{=} \sup_{\xi \in [0,1]} \phi(x, \xi),$$

with $\phi : \mathbb{R}^n \times [0, 1] \to \mathbb{R}$ continuously differentiable in the first argument. Such problems arise in numerous application areas, such as engineering design, where a specification must be satisfied over a range of independent parameter values. For an excellent survey of the theory behind the problem (SI), in addition to some algorithms and applications, see [27] as well as the collection [67][1]. Many

[1] Some of the content of this chapter has appeared in the article [35] in the collection [67].

globally convergent algorithms designed to solve (SI) rely on approximating $\Phi(x)$ by using progressively finer discretizations of $[0,1]$ (see, e.g. [18, 21, 26, 49, 54, 55, 58, 65]). Specifically, such algorithms generate a sequence of problems of the form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & \phi(x,\xi) \le 0, \quad \forall \xi \in \Xi, \end{array} \tag{DSI}$$

where $\Xi \subset [0,1]$ is a (presumably large) finite set. For example, given $q \in \mathbb{N}$, one could use the *uniform discretization*

$$\Xi \triangleq \left\{ 0, \frac{1}{q}, \ldots, \frac{q-1}{q}, 1 \right\}.$$

Clearly these algorithms are crucially dependent upon being able to efficiently solve problem (DSI).

Of course, (DSI) involves only a finite number of smooth constraints, thus could be solved in principle via a standard constrained optimization algorithm such as that introduced in Chapter 3. Note however that when $|\Xi|$ is large compared to the number of variables n, it is likely that only a small subset of the constraints are active at a solution. A scheme which exploits this fact by cleverly using an appropriate small subset of the constraints at each step should, in most cases, enjoy substantial savings in computational effort without sacrificing global and local convergence properties.

Early efforts at employing such a scheme appear in [55, 49] in the context of first order methods of feasible directions. In [55], at iteration k, a search direction is computed based on the method of Zoutendijk [80] using only the gradients of those constraints satisfying $\phi(x_k,\xi) \ge -\epsilon$, where $\epsilon > 0$ is small. Clearly, close to a solution, such "ϵ-active" constraints are sufficient to ensure convergence. However, if the discretization is very fine, such an approach may still produce sub-problems with an unduly large number of constraints. It was shown in [49] that, by means of a scheme inspired by the bundle-type methods of non-differentiable optimization (see, e.g. [33, 37]), the number of constraints used in the sub-problems can be further reduced without jeopardizing global convergence. Specifically, in [49], the constraints to be used in the computation of the search direction d_{k+1} at iteration $k+1$ are chosen as follows. Let $\Xi_k \subseteq \Xi$ be the set of constraints used to compute the search direction d_k, and let x_{k+1} be the next iterate. Then Ξ_{k+1} includes:

- All $\xi \in \Xi$ such that $\phi(x_{k+1}, \xi) = 0$ (i.e. the "active" constraints),
- All $\xi \in \Xi_k$ which influenced the computation of the search direction d_k, and
- Some $\xi \in \Xi$, if it exists, which caused a step-length reduction in the line search at iteration k.

While the former is needed to ensure that d_k is a feasible direction, it is argued in [49] that the latter two are necessary to avoid zig-zagging or other jamming phenomena. The number of constraints required to compute the search direction is thus typically small compared to $|\Xi|$, hence each iteration of such a method is computationally less costly. Unfortunately, for a fixed level of discretization, the algorithms in [55, 49] converge at a linear rate at best.

A number of attempts at applying the SQP scheme to problems with a large number of constraints, e.g. our discretized problem from SIP, have been documented in the literature. In [2], Biggs treats all active inequality constraints as equality constraints in the QP sub-problem, while ignoring all constraints which are not active. Polak and Tits [58], and Mine et al. [43], apply to the SQP framework an ϵ-active scheme similar to that used in [55]. Similar to the ϵ-active idea, Powell proposes a "tolerant" algorithm for linearly constrained problems in [62]. Finally, in [71], Schittkowski proposes another modification of the SQP scheme for problems with many constraints, but does not prove convergence. In practice, the algorithm in [71] may or may not converge, dependent upon the heuristics applied to choose the constraints for the QP sub-problem.

In this chapter, the scheme introduced in [49] in the context of first-order feasible direction methods is extended to the Feasible SQP (FSQP) framework introduced in Chapter 3. Our presentation and analysis significantly borrow from [35], where the FSQP algorithm of [51] is similarly extended to handle problems with many constraints. The algorithm and analysis of [35] were inspired by that of [78], where an important special case of (DSI) is considered, the unconstrained minimax problem.

Let the feasible set be denoted

$$X \stackrel{\Delta}{=} \{x \in \mathbb{R}^n \mid \phi(x,\xi) \leq 0, \ \forall \xi \in \Xi \ \}.$$

For $x \in X$, $\eta \geq 0$, $\hat{\Xi} \subseteq \Xi$, and $H \in \mathbb{R}^{n \times n}$ with $H = H^T > 0$, let

$$(\hat{d}(x,H,\eta,\Xi'), \hat{\gamma}(x,H,\eta,\Xi')) \in \mathbb{R}^n \times \mathbb{R}$$

be the unique solution of the QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle \hat{d}, H\hat{d}\rangle + \hat{\gamma} \\ \text{s.t.} & \langle \nabla f(x), \hat{d}\rangle \leq \hat{\gamma} \\ & \phi(x,\xi) + \langle \nabla_x \phi(x,\xi), \hat{d}\rangle \leq \hat{\gamma}\cdot\eta, \quad \forall \xi \in \Xi'. \end{array} \qquad \widehat{QP}(x,H,\eta,\Xi')$$

At iteration k, given an estimate $x_k \in X$ of the solution, a constraint index set $\Xi_k \subseteq \Xi$, $\eta_k > 0$, and a symmetric positive definite estimate H_k of the Hessian of the Lagrangian, the basic algorithm presented in this chapter first computes $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k, \Xi_k)$. An Armijo-type line search is then performed along the direction $\hat{d}_k$, yielding a step-size $t_k \in (0, 1]$. The next iterate is taken to be $x_{k+1} = x_k + t_k\hat{d}_k$. Finally, H_k is updated yielding H_{k+1}, the tilting parameter η_k is updated to η_{k+1}, and a new constraint index set Ξ_{k+1} is constructed following the ideas of [49].

As is pointed out in [78], the construction of [49] cannot be used meaningfully in the SQP framework without modifying the update rule for the new metric H_{k+1}. The reason is as follows. As discussed above, following [49], if $t_k < 1$, Ξ_{k+1} is to include, among others, the index $\bar{\xi} \in \Xi$ of a constraint which was infeasible for the last trial point in the line search.[2] The rationale for including $\bar{\xi}$ in Ξ_{k+1} is that if $\bar{\xi}$ had been in Ξ_k, then it is likely that the computed search direction would have allowed a longer step. Such reasoning is clearly justified in the context of first-order search directions as is used in [49], but it is not clear that $\bar{\xi}$ is the right constraint to include under the new metric H_{k+1}. To overcome this difficulty, it is proposed in [78] that H_k not be updated whenever $t_k < \delta$, δ a prescribed small positive number, and $\bar{\xi} \notin \Xi_k$. We will show in Section 5.3 that, as is the case for the mini-max algorithm of [78], for k large enough, $\bar{\xi}$ will always be in Ξ_k, thus normal updating will take place eventually, preserving the local convergence rate properties of the SQP scheme.

As a final matter on the update rule for Ξ_k, following [78], we allow for additional constraint indices to be added to the set Ξ_k. While not necessary for global convergence, cleverly choosing additional constraints can significantly

[2] Assuming that it was a constraint, and not the objective function, which caused a failure in the line search.

improve performance, especially in early iterations. In the context of discretized SIP, exploiting the possible regularity properties of the SIP constraints with respect to the independent parameter can give useful heuristics for choosing additional constraints.

In order to guarantee fast (superlinear) local convergence, it is again necessary that, for k large enough, the line search always accept the step-size $t_k = 1$. It is well-known in the SQP framework that the line search could truncate the step size arbitrarily close to a solution (the so-called Maratos effect discussed in Section 2.3), thus preventing superlinear convergence. Various schemes have been devised to overcome such a situation. We will argue that a second-order correction, as used in Chapter 3, will still be sufficient to overcome the Maratos effect without sacrificing global convergence.

The balance of the chapter is organized as follows. In Section 5.2 we introduce the algorithm and present some preliminary material. Next, in Section 5.3, we give a complete convergence analysis of the algorithm proposed in Section 5.2. The local convergence analysis assumes the just mentioned second-order correction is used. In Section 5.4, the algorithm is extended to handle the constrained mini-max case. Some implementation details, in addition to numerical results, are provided in Section 5.5.

5.2 Algorithm

We begin by making a few assumptions that will be in force throughout. They are the same as those that were used in Chapter 3. The first is a standard regularity assumption, while the second ensures that the set of active constraint gradients is always linearly independent.

Assumption 1: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $\phi(\cdot, \xi) : \mathbb{R}^n \to \mathbb{R}$, $\xi \in \Xi$, are continuously differentiable.

Define the set of active constraints for a point $x \in X$ as

$$\Xi_{\text{act}}(x) \stackrel{\Delta}{=} \{\xi \in \Xi \mid \phi(x, \xi) = 0\}.$$

Assumption 2: For all $x \in X$ with $\Xi_{\text{act}}(x) \neq \emptyset$, the set $\{\nabla_x \phi(x, \xi) \mid \xi \in \Xi_{\text{act}}(x)\}$ is linearly independent.

Applying the definition given in Section 2.1 to (DSI), a point $x^* \in \mathbb{R}^n$ is called a *Karush-Kuhn-Tucker (KKT)* point for the problem (DSI) if there exist KKT multipliers $\lambda^{*,\xi}$, $\xi \in \Xi$, satisfying

$$\begin{cases} \nabla f(x^*) + \displaystyle\sum_{\xi \in \Xi} \lambda^{*,\xi} \nabla_x \phi(x^*, \xi) = 0, \\ \phi(x^*, \xi) \leq 0, \quad \forall \xi \in \Xi, \\ \lambda^{*,\xi} \phi(x^*, \xi) = 0 \text{ and } \lambda^{*,\xi} \geq 0, \quad \forall \xi \in \Xi. \end{cases} \tag{5.1}$$

Under our assumptions, any local minimizer x^* for (DSI) is a KKT point. Thus, (5.1) provides a set of first-order necessary conditions of optimality.

Throughout our analysis, we will often refer to the KKT conditions for $\widehat{QP}(x, H, \eta, \Xi')$. Specifically, given $x \in X$, $H = H^T > 0$, $\eta \geq 0$, and $\Xi' \subseteq \Xi$, $(\hat{d}, \hat{\gamma})$ is a KKT point for $\widehat{QP}(x, H, \eta, \Xi')$ if there exist (scalar) multipliers $\hat{\mu}$ and

$\hat{\lambda}^{\xi}$, $\xi \in \Xi'$, satisfying

$$\begin{cases} \begin{bmatrix} H\hat{d} \\ 1 \end{bmatrix} + \hat{\mu} \begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix} + \sum_{\xi \in \Xi'} \hat{\lambda}^{\xi} \begin{bmatrix} \nabla_x \phi(x,\xi) \\ -\eta \end{bmatrix} = 0, \\ \langle \nabla f(x), \hat{d} \rangle \leq \hat{\gamma}, \\ \phi(x,\xi) + \langle \nabla_x \phi(x,\xi), \hat{d} \rangle \leq \hat{\gamma} \cdot \eta, \quad \forall \xi \in \Xi', \\ \hat{\mu} \left(\langle \nabla f(x), \hat{d} \rangle - \hat{\gamma} \right) = 0 \text{ and } \hat{\mu} \geq 0, \\ \hat{\lambda}^{\xi} \left(\phi(x,\xi) + \langle \nabla_x \phi(x,\xi), \hat{d} \rangle - \hat{\gamma} \cdot \eta \right) = 0 \text{ and } \hat{\lambda}^{\xi} \geq 0, \quad \forall \xi \in \Xi'. \end{cases} \tag{5.2}$$

In fact, such a $(\hat{d}, \hat{\gamma})$ is the unique KKT point, as well as the unique global minimizer (stated formally in Lemma 43 below). As in Chapter 3, we will make frequent use of a simple consequence of the first equation in (5.2), i.e.

$$\hat{\mu} + \eta \cdot \sum_{\xi \in \Xi'} \hat{\lambda}^{\xi} = 1. \tag{5.3}$$

It remains to explicitly specify the key feature of the proposed algorithm: the update rule for Ξ_k. As discussed in Section 5.1, following [49], Ξ_{k+1} will include (in addition to possible heuristics) three crucial components. The first is the set $\Xi_{\text{act}}(x_{k+1})$ of indices of active constraints at the new iterate. The second component of Ξ_{k+1} is the set $\hat{\Xi}_k^b \subseteq \Xi_k$ of indices of constraints that affected $\hat{d}_k$. In particular, $\hat{\Xi}_k^b$ will include all indices of constraints in $\widehat{QP}(x_k, H_k, \eta_k, \Xi_k)$ which have positive multipliers, i.e. the *binding* constraints. Specifically, let $\hat{\lambda}_k^{\xi}$, $\xi \in \Xi_k$, be the QP multipliers from $\widehat{QP}(x_k, H_k, \eta_k, \Xi_k)$. Define

$$\hat{\Xi}_k^b \triangleq \{ \ \xi \in \Xi_k \mid \hat{\lambda}_k^{\xi} > 0 \ \}.$$

Finally, the third component of Ξ_{k+1} is the index $\bar{\xi}$ of one constraint, if any exists, which forced a reduction of the step in the previous line search. While the exact type of line search we employ is not critical to our analysis, we assume from this point onward that it is an Armijo-type search (as was used in Chapter 3). That is, given constants $\alpha \in (0, 1/2)$ and $\beta \in (0, 1)$, the step-size t_k is taken as the first number t in the set $\{1, \beta, \beta^2, \dots\}$ such that

$$f(x_k + td_k) \leq f(x_k) + \alpha t \langle \nabla f(x_k), \hat{d}_k \rangle, \tag{5.4}$$

and

$$\phi(x_k + t\hat{d}_k, \xi) \leq 0, \quad \forall \xi \in \Xi. \tag{5.5}$$

Thus, $t_k < 1$ implies that either (5.4) or (5.5) is violated at $x_k + \frac{t_k}{\beta}\hat{d}_k$. In the event that (5.5) is violated, there exists $\bar{\xi} \in \Xi$ such that

$$\phi\left(x_k + \frac{t_k}{\beta}\hat{d}_k,\ \bar{\xi}\right) > 0, \tag{5.6}$$

and, in such a case, we will include $\bar{\xi}$ in Ξ_{k+1}.

In order to update the tilting parameter, we follow the same scheme as in Chapter 3. Specifically, given an index set $\Xi''_k \subseteq \Xi$, we attempt to compute an estimate $\widehat{d^0_k} = \widehat{d^0}(x_k, H_k, \Xi''_k)$ of the SQP direction by solving the equality constrained QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle \widehat{d^0}, H_k \widehat{d^0} \rangle + \langle \nabla f(x_k), \widehat{d^0} \rangle \\ \text{s.t.} & \phi(x_k, \xi) + \langle \nabla_x \phi(x_k, \xi), \widehat{d^0} \rangle = 0, \quad \xi \in \Xi''_k, \end{array} \qquad LS^0(x_k, H_k, \Xi''_k)$$

which, again, is equivalent (after a change of variables) to solving a linear least squares problem. Define

$$\hat{\Xi}_k \triangleq \{\ \xi \in \Xi_k \mid \phi(x_k, \xi) + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle = \hat{\gamma}_k \cdot \eta_k\ \}.$$

We will show in Section 5.3.2 that, in order to guarantee fast local convergence, it is sufficient to choose

$$\Xi''_k = \hat{\Xi}_{k-1}.$$

If $\widehat{d_k^0}$ exists, is bounded, and has positive multipliers (i.e. is a "good" estimate), the tilting parameter is taken as

$$\eta_k = C_k \cdot \|\widehat{d_k^0}\|^2,$$

where $C_k > 0$ is chosen according to heuristics. Otherwise, it is sufficient to use $\min\{\epsilon_\ell, \|\hat{d}_{k-1}\|^2\}$ in place of $\|\widehat{d_k^0}\|^2$.

Algorithm FSQP′-MC

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $0 < \delta \ll 1$, $\epsilon_\ell > 0$, $0 < \underline{C} \leq \overline{C}$, $\bar{D} > 0$.

Data. $x_0 \in X$, $0 < H_0 = H_0^T \in \mathbb{R}^{n \times n}$.

Step 0 - Initialization. **set** $k \leftarrow 0$ and **choose** $\Xi_0 \supseteq \Xi_{\text{act}}(x_0)$.

Step 1 - Computation of search direction.

(i) **compute** $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k, \Xi_k)$.

(ii) **if** $\hat{d}_k = 0$, **then stop**.

Step 2 - Line search. **compute** t_k, the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying (5.4) and (5.5).

Step 3 - Updates.

(i). **set** $x_{k+1} \leftarrow x_k + t_k \hat{d}_k$.

(ii). **if** $t_k < 1$ and (5.5) was violated at $x_k + \frac{t_k}{\beta}\hat{d}_k$, **then** let $\bar{\xi}$ be such that (5.6) holds.

(iii). **pick**

$$\Xi_{k+1} \supseteq \Xi_{\text{act}}(x_{k+1}) \cup \hat{\Xi}_k^b.$$

```
    if $t_k < 1$ and (5.6) holds for some $\bar{\xi} \in \Xi$, then set
            $\Xi_{k+1} \leftarrow \Xi_{k+1} \cup \{\bar{\xi}\}$.
(iv). if $t_k \leq \delta$ and $\bar{\xi} \notin \Xi_k$, then set $H_{k+1} \leftarrow H_k$, $\eta_{k+1} \leftarrow \eta_k$. else,
    (a) compute a new symmetric positive definite estimate
    $H_{k+1}$ to the Hessian of the Lagrangian.
    (b) select $C_{k+1} \in [\underline{C}, \overline{C}]$.
  * if ($\|\hat{d}_k\| < \epsilon_\ell$) then
      · compute, if possible,[3] $\widehat{d^0_{k+1}} = \widehat{d^0}(x_{k+1}, H_{k+1}, \hat{\Xi}_k)$, and the
        associated multipliers $\widehat{\lambda^0_{k+1}} \in \mathbb{R}^{|\hat{\Xi}_k|}$.
      · if $\left(\widehat{d^0_{k+1}}\right.$ exists and $\|\widehat{d^0_{k+1}}\| \leq \bar{D}$ and $\left.\widehat{\lambda^0_{k+1}} \geq 0\right)$ then set
            $\eta_{k+1} \leftarrow C_{k+1} \cdot \|\widehat{d^0_{k+1}}\|^2$.
      · else set $\eta_{k+1} \leftarrow C_{k+1} \cdot \|\hat{d}_k\|^2$.
  * else set $\eta_{k+1} \leftarrow C_{k+1} \cdot \epsilon_\ell^2$.
(v). set $k \leftarrow k + 1$ and goto Step 1.
```

5.3 Convergence Analysis

While there are some critical differences, the analysis in this section closely parallels that of [78, 35]. We begin by establishing that, under a few additional assumptions, algorithm **FSQP′-MC** generates a sequence which converges to a KKT point for (DSI). Then, upon strengthening our assumptions slightly, we show that the rate of convergence is two-step superlinear.

[3] That is, if $LS^0(x_{k+1}, H_{k+1}, \hat{\Xi}_k)$ is non-degenerate.

5.3.1 Global Convergence

The following will be assumed to hold throughout our analysis.

Assumption 3: The level set $\{ \ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \ \} \cap X$ is compact.

Assumption 4: There exist scalars $0 < \sigma_1 \leq \sigma_2$ such that for all k,

$$\sigma_1 \|d\|^2 \leq \langle d, H_k d \rangle \leq \sigma_2 \|d\|^2, \quad \forall d \in \mathbb{R}^n.$$

Given the scalars $0 < \sigma_1 \leq \sigma_2$ from Assumption 4, define

$$\mathcal{H} \overset{\Delta}{=} \{H = H^T \mid \sigma_1 \|d\|^2 \leq \langle d, Hd \rangle \leq \sigma_2 \|d\|^2, \quad \forall d \in \mathbb{R}^n\}.$$

First, we derive some properties of $\hat{d}(x, H, \eta, \Xi')$.

Lemma 43. *For all $x \in X$, $H \in \mathcal{H}$, $\eta \geq 0$, and $\Xi' \subseteq \Xi$, the pair*

$$(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta, \Xi'), \hat{\gamma}(x, H, \eta, \Xi')) \in \mathbb{R}^n \times \mathbb{R}$$

is well-defined and the unique KKT point of $\widehat{QP}(x, H, \eta, \Xi')$. Further, for $\Xi' \subseteq \Xi$ fixed, suppose $\{x_k\}_{k \in \mathbb{N}} \subset X$ is bounded, $\{H_k\}_{k \in \mathbb{N}} \subset \mathcal{H}$, and $\{\eta_k\}_{k \in \mathbb{N}} \subset [0, \infty)$. Then $\{\hat{d}(x_k, H_k, \eta_k, \Xi')\}_{k \in \mathbb{N}}$ is bounded. Finally, $\hat{d} = 0$ if, and only if, $\hat{\gamma} = 0$.

Proof. The first and second claims are proved exactly as in Lemma 1. The third claim is proved exactly as in Lemma 2. □

The following results are straightforward extensions of Lemmas 2, 3, and 4 in Chapter 3. The proofs are omitted here as they are minor modifications of those given in Chapter 3.

Lemma 44. *For all $x \in X$, $H \in \mathcal{H}$, $\eta \geq 0$, and $\Xi' \subseteq \Xi$ such that $\Xi_{act}(x) \subseteq \Xi'$, $(\hat{d}(x, H, \eta, \Xi'), \hat{\gamma}(x, H, \eta, \Xi')) = (0, 0)$ if, and only if, x is a KKT point for (DSI). If x is not a KKT point for (DSI) and $\eta > 0$, then $\hat{d} = \hat{d}(x, H, \eta, \Xi')$ satisfies*

$$\langle \nabla f(x), \hat{d} \rangle < 0, \tag{5.7}$$

$$\langle \nabla_x \phi(x, \xi), \hat{d} \rangle < 0, \quad \forall \xi \in \Xi_{act}(x), \tag{5.8}$$

and $\hat{\gamma} = \hat{\gamma}(x, H, \eta, \Xi') < 0$.

Lemma 45. *If $\eta_k = 0$, then x_k is KKT for (DSI) and the algorithm will stop in Step 1(ii) at iteration k. On the other hand, whenever the algorithm does not stop in Step 1(ii), the line search is well-defined, i.e. Step 2 yields a step $t_k = \beta^j$ for some finite $j = j(k)$.*

In view of the update rule in *Step 3(iii)* and Lemma 44, if Algorithm **FSQP′-MC** generates a finite sequence terminating at the point x_N, then x_N is a KKT point for (DSI). We now concentrate on the case in which the algorithm never satisfies the termination condition in *Step 1(ii)* and generates an infinite sequence $\{x_k\}$. As a consequence of Lemma 45, we may assume throughout that

$$\eta_k > 0, \quad \forall k \in \mathbb{N}. \tag{5.9}$$

Before stating the next lemma, recall that a set of vectors $\{ \ v_j \in \mathbb{R}^n \mid j = 1, \dots, r \ \}$ is said to be *positive linear independent* if there does not exist scalars $\alpha_j \geq 0$, $j = 1, \dots, r$, not all zero, such that

$$\sum_{j=1}^{r} \alpha_j v_j = 0.$$

Lemma 46. *Suppose $\mathcal{K}$ is an infinite index set such that $\hat{\Xi}_k \equiv \Xi'$, for all $k \in \mathcal{K}$, $x_k \xrightarrow{k\in\mathcal{K}} x^* \in X$, $H_k \xrightarrow{k\in\mathcal{K}} H^* \in \mathcal{H}$, $\eta_k \xrightarrow{k\in\mathcal{K}} 0$, $\hat{d}_k \xrightarrow{k\in\mathcal{K}} \hat{d}^*$, and $\hat{\gamma}_k$ is bounded. Then the set*

$$\{ \nabla_x \phi(x^*, \xi) \mid \xi \in \Xi' \}$$

is positive linear independent.

Proof. We argue by contradiction. Suppose there exist scalars $\alpha_\xi \geq 0$, $\xi \in \Xi'$, not all zero, such that

$$\sum_{\xi \in \Xi'} \alpha_\xi \nabla_x \phi(x^*, \xi) = 0. \tag{5.10}$$

Taking the limit on $\mathcal{K}$ in the QP-active constraints (since $\eta_k \xrightarrow{k\in\mathcal{K}} 0$ and $\hat{\gamma}_k$ is bounded)

$$\phi(x^*, \xi) + \langle \nabla_x \phi(x^*, \xi), \hat{d}^* \rangle = 0, \quad \forall \xi \in \Xi'. \tag{5.11}$$

Taking the inner product of (5.10) with $\hat{d}^*$ and substituting (5.11) gives

$$-\sum_{\xi \in \Xi'} \alpha_\xi \phi(x^*, \xi) = 0.$$

Thus, since $x^* \in X$, we must have

$$\phi(x^*, \xi) = 0, \quad \forall \xi \in \{ \xi \mid \alpha_\xi > 0 \},$$

i.e. $\{ \xi \mid \alpha_\xi > 0 \} \subseteq \Xi_{\text{act}}(x^*)$, in which case (5.10) contradicts Assumption 2. □

Lemma 47. *Suppose $\mathcal{K}$ is an infinite index set such that $\hat{\Xi}_k \equiv \Xi^*$, for all $k \in \mathcal{K}$, $x_k \xrightarrow{k\in\mathcal{K}} x^* \in X$, $H_k \xrightarrow{k\in\mathcal{K}} H^* \in \mathcal{H}$, $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \geq 0$, $\hat{d}_k \xrightarrow{k\in\mathcal{K}} \hat{d}^*$, and $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} \hat{\gamma}^*$. Then $(\hat{d}^*, \hat{\gamma}^*)$ is the unique KKT point of $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$.*

Proof. We begin by showing that $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is bounded. It is clear from positivity of the multipliers and (5.3) that $\hat{\mu}_k \in [0,1]$ for all k, hence $\{\hat{\mu}_k\}_{k\in\mathcal{K}}$ is bounded. First, consider the case that $\eta^* > 0$. From (5.3) we have

$$\sum_{\xi\in\Xi^*} \hat{\lambda}_k^\xi = \frac{1}{\eta_k}(1-\hat{\mu}_k).$$

Since $\eta^* > 0$, η_k is bounded away from zero on $\mathcal{K}$. As $\hat{\lambda}_k^\xi \geq 0$, for all $k \in \mathcal{K}$, $\xi \in \Xi^*$, and $\{\hat{\mu}_k\}_{k\in\mathcal{K}}$ is bounded, it immediately follows that $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is bounded.

Now consider the case that $\eta^* = 0$ and, proceeding by contradiction, suppose that $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is unbounded. Let $\mathcal{K}' \subseteq \mathcal{K}$ be an infinite index set such that $\|\hat{\lambda}_k\| \xrightarrow{k\in\mathcal{K}'} \infty$. Define

$$\nu_k^\xi \triangleq \frac{\hat{\lambda}_k^\xi}{\|\hat{\lambda}_k\|}, \quad \xi \in \Xi^*,$$

and suppose without loss of generality that $\nu_k^\xi \xrightarrow{k\in\mathcal{K}'} \nu^{*,\xi} \in [0,1]$, $\xi \in \Xi^*$. Of course, $\|\nu_k\| = 1$, for all $k \in \mathcal{K}'$, thus $\|\nu^*\| = 1$. Divide the first equation of the QP optimality conditions (5.2) by $\|\hat{\lambda}_k\|$ and take the limit on $\mathcal{K}'$ (all quantities are convergent on $\mathcal{K}'$), yielding

$$\sum_{\xi\in\Xi^*} \nu^{*,\xi}\nabla_x\phi(x^*,\xi) = 0.$$

Note that $\nu^{*,\xi} > 0$ implies $\xi \in \hat{\Xi}_k$, for all k sufficiently large. We assume without loss of generality that $\hat{\Xi}_k \equiv \hat{\Xi}$ for all $k \in \mathcal{K}'$. In view of Lemma 46, since $\|\nu^*\| = 1$, we have a contradiction. Therefore, $\{\hat{\lambda}_k\}_{k\in\mathcal{K}}$ is bounded.

Now suppose that $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite index set such that $\hat{\mu}_k \xrightarrow{k\in\mathcal{K}'} \hat{\mu}^*$, and $\hat{\lambda}_k^\xi \xrightarrow{k\in\mathcal{K}'} \hat{\lambda}^{*,\xi}$, $\xi \in \Xi^*$. Taking limits in the optimality conditions (5.2) shows that $(\hat{d}^*, \hat{\gamma}^*)$ is a KKT point for $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$ with multipliers $\hat{\mu}^* \geq 0$ and $\hat{\lambda}^{*,\xi} \geq 0$, $\xi \in \Xi^*$. Uniqueness of such points (Lemma 43) proves the result. □

Lemma 48. *(i) The sequences $\{x_k\}$, $\{\eta_k\}$, and $\{\hat{d}_k\}$ are bounded; (ii) $\{f(x_k)\}$ converges; (iii) $t_k d_k \longrightarrow 0$.*

Proof. Boundedness of $\{x_k\}$ follows from Assumption 3 and the fact that $\{f(x_k)\}$ is a monotonically decreasing sequence (guaranteed by *Step 2*). Since f is continuous, it also follows that $\{f(x_k)\}$ converges. It follows from *Step 3(iv)* that $\{\eta_k\}$ is bounded, thus in view of Assumption 4, Lemma 43, and boundedness of $\{x_k\}$, $\{\hat{d}_k\}$ is bounded.

In view of the objective descent constraint in $\widehat{QP}(x_k, H_k, \eta_k, \Xi_k)$ and since $(0,0)$ is always feasible,

$$\begin{array}{rcl} \langle \nabla f(x_k), \hat{d}_k \rangle & \leq & \hat{\gamma}_k \\ & \leq & -\tfrac{1}{2}\langle \hat{d}_k, H_k \hat{d}_k \rangle. \end{array}$$

From *Step 2*, we have

$$\begin{array}{rcl} f(x_{k+1}) & \leq & f(x_k) + \alpha t_k \langle \nabla f(x_k), \hat{d}_k \rangle \\ & \leq & f(x_k) - \tfrac{\alpha}{2} t_k \langle \hat{d}_k, H_k \hat{d}_k \rangle. \end{array}$$

Rearranging,

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha}{2} t_k \langle \hat{d}_k, H_k \hat{d}_k \rangle \geq 0.$$

In view of the second claim of this lemma, and since $t_k \in [0,1]$, for all k, we conclude

$$\langle t_k \hat{d}_k, H_k (t_k \hat{d}_k) \rangle \to 0.$$

As $H_k \in \mathcal{H}$ for all k, claim (*iii*) follows. □

In order to establish convergence to a KKT point, it will be convenient to consider the value function for $\widehat{QP}(x, H, \eta, \Xi')$. In particular, given the solution $(\hat{d}, \hat{\gamma}) = (\hat{d}(x, H, \eta, \Xi'), \hat{\gamma}(x, H, \eta, \Xi'))$, define

$$\hat{v}(x, H, \eta, \Xi') \triangleq -\left(\frac{1}{2}\langle \hat{d}, H\hat{d} \rangle + \hat{\gamma}\right).$$

Note that, since $(0,0)$ is always feasible for $\widehat{QP}(x, H, \eta, \Xi')$,

$$\hat{v}_k \triangleq \hat{v}(x_k, H_k, \eta_k, \Xi_k) \geq 0,$$

for all k.

Lemma 49. *Let $\mathcal{K}$ be an infinite index set. Then (i) $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$ if and only if $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} 0$, (ii) $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$ if and only if $\hat{v}_k \xrightarrow{k\in\mathcal{K}} 0$, and (iii) if $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$, then all accumulation points of $\{x_k\}_{k\in\mathcal{K}}$ are KKT points for (DSI).*

Proof. Suppose $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$ and assume without loss of generality that $x_k \xrightarrow{k\in\mathcal{K}} x^* \in X$, $H_k \xrightarrow{k\in\mathcal{K}} H^* \in \mathcal{H}$, $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \geq 0$, $\Xi_k \equiv \Xi'$ for all $k \in \mathcal{K}$, and $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} \hat{\gamma}^*$. Then, in view of Lemma 47, $(0, \hat{\gamma}^*)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*, \Xi')$. It follows from Lemma 43 that $\hat{\gamma}^* = 0$. The converse is proved similarly, hence claim (i) is proved.

Now suppose that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Then $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}} 0$ and it is clear from the definition of $\hat{v}_k$ that $\hat{v}_k \xrightarrow{k\in\mathcal{K}} 0$. To prove the converse, note that from the optimality conditions (5.2),

$$\begin{aligned}
\langle \hat{d}_k, H_k \hat{d}_k \rangle &= -\hat{\mu}_k \langle \nabla f(x_k), \hat{d}_k \rangle - \sum_{\xi \in \Xi_k} \hat{\lambda}_k^\xi \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \\
&= -\hat{\mu}_k \hat{\gamma}_k - \sum_{\xi \in \Xi_k} \hat{\lambda}_k^\xi (\hat{\gamma}_k \cdot \eta_k - \phi(x_k, \xi)) \\
&= \left(\hat{\mu}_k + \eta_k \cdot \sum_{\xi \in \Xi_k} \hat{\lambda}_k^\xi \right) \hat{\gamma}_k + \sum_{\xi \in \Xi_k} \hat{\lambda}_k^\xi \phi(x_k, \xi) \\
&\leq -\hat{\gamma}_k.
\end{aligned}$$

Thus,

$$\begin{aligned}
\hat{v}_k &= -\frac{1}{2} \langle \hat{d}_k, H_k \hat{d}_k \rangle - \hat{\gamma}_k \\
&\geq \frac{1}{2} \langle \hat{d}_k, H_k \hat{d}_k \rangle > 0,
\end{aligned}$$

for all $k \in \mathcal{K}$. In view of Assumption 4, it is clear that if $\hat{v}_k \xrightarrow{k\in\mathcal{K}} 0$, then $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Thus, claim (i) is proved.

Suppose that $\hat{d}_k \xrightarrow{k\in\mathcal{K}} 0$. Let x^* be some accumulation point of $\{x_k\}$ and suppose that $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}'} x^*$. Without loss of generality, assume that $\Xi_k \equiv \Xi'$ for all $k \in \mathcal{K}'$, $H_k \xrightarrow{k\in\mathcal{K}'} H^*$, and $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^* \geq 0$. As usual, let $\hat{\mu}_k$ and $\hat{\lambda}_k^\xi$, $\xi \in \Xi'$ denote the multipliers from $\widehat{QP}(x_k, H_k, \eta_k, \Xi')$ and suppose further, without loss of generality, that $\hat{\Xi}_k \equiv \hat{\Xi}'$, for all $k \in \mathcal{K}$. As $\hat{d}_k \xrightarrow{k\in\mathcal{K}'} 0$, $\hat{\gamma}_k \xrightarrow{k\in\mathcal{K}'} 0$ and it is clear that $\hat{\Xi}' \subseteq \Xi_{\text{act}}(x^*)$. Thus, in view of Assumption 2,

$$\{\ \nabla_x \phi(x_k, \xi) \mid \xi \in \hat{\Xi}'\ \}$$

is a linearly independent set for all $k \in \mathcal{K}'$, k sufficiently large.

Again, without loss of generality, suppose that $\hat{\mu}_k \xrightarrow{k\in\mathcal{K}'} \hat{\mu}^* \in [0,1]$. Define

$$\hat{R}(x) \stackrel{\Delta}{=} [\ \nabla_x \phi(x, \xi) \mid \xi \in \hat{\Xi}'\ ],$$

and let $\hat{R}_k \stackrel{\Delta}{=} \hat{R}(x_k)$. From the optimality conditions (5.2) we obtain the unique expression for the multipliers

$$\hat{\lambda}_k = -\left(\hat{R}_k^T \hat{R}_k\right)^{-1} \hat{R}_k^T \left(H_k \hat{d}_k + \hat{\mu}_k \nabla f(x_k)\right).$$

In view of Assumptions 1 and 4, boundedness of $\{x_k\}$, and since $\hat{d}_k \xrightarrow{k\in\mathcal{K}'} 0$, we see

$$\hat{\lambda}_k \xrightarrow{k\in\mathcal{K}'} \hat{\lambda}^* = -\left(\hat{R}^{*T} \hat{R}^*\right)^{-1} \hat{R}^{*T} \left(\hat{\mu}^* \nabla f(x^*)\right),$$

where $\hat{R}^* = \hat{R}(x^*)$. Taking limits in the optimality conditions (5.2) shows that $\hat{\mu}^*$ and $\hat{\lambda}^*$ are multipliers for $\widehat{QP}(x^*, H^*, \eta^*, \hat{\Xi}')$, where we set $\hat{\lambda}^{*,\xi} = 0$ for $\xi \in \Xi' \backslash \hat{\Xi}'$. Note that, from (5.3) and our explicit expression for $\hat{\lambda}^*$ above, $\hat{\mu}^* > 0$. Finally, it is not difficult to see from (5.2) that x^* is KKT for (DSI) with multipliers

$$\lambda^{*,\xi} = \begin{cases} \dfrac{\hat{\lambda}^{*,\xi}}{\hat{\mu}^*}, & \xi \in \Xi', \\ 0, & \text{otherwise.} \end{cases}$$

□

Lemma 50. *Given $x \in X$, $H > 0$, and $\eta \geq 0$, suppose $\Xi' \subset \Xi'' \subseteq \Xi$. If $\hat{d}(x, H, \eta, \Xi')$ is not feasible for $\widehat{QP}(x, H, \eta, \Xi'')$, then*

$$\hat{v}(x, H, \eta, \Xi'') < \hat{v}(x, H, \eta, \Xi').$$

Proof. Clearly $\hat{d}(x, H, \eta, \Xi') \neq \hat{d}(x, H, \eta, \Xi'')$ since, by assumption, $\hat{d}(x, H, \eta, \Xi')$ is not feasible for $\widehat{QP}(x, H, \eta, \Xi'')$. On the other hand, $\hat{d}(x, H, \eta, \Xi'')$ is feasible for $\widehat{QP}(x, H, \eta, \Xi')$. Uniqueness of the solution of $\widehat{QP}(x, H, \eta, \Xi')$ (Lemma 43) gives the result. □

The proof of the following two results were inspired by the proof of Theorem T in [49].

Lemma 51. *Suppose $\mathcal{K}$ is an infinite index set such that*

$$\begin{aligned} &x_k \xrightarrow{k \in \mathcal{K}} x^* \in X, \quad H_k \xrightarrow{k \in \mathcal{K}} H^* \in \mathcal{H}, \quad \eta_k \xrightarrow{k \in \mathcal{K}} \eta^* \geq 0, \\ &\hat{d}_k \xrightarrow{k \in \mathcal{K}} \hat{d}^*, \qquad\quad \hat{\gamma}_k \xrightarrow{k \in \mathcal{K}} \hat{\gamma}^*, \end{aligned}$$

where x^* is not a KKT point for (DSI), and suppose $\Xi_k \equiv \Xi'$ for all $k \in \mathcal{K}$. Then $\eta^* > 0$ and there exists $\underline{t} > 0$ such that for all $t \in [0, \underline{t}]$, $\phi(x_k + td_k, \xi) \leq 0$, for all $\xi \in \Xi'$, and for all $k \in \mathcal{K}$ sufficiently large.

Proof. We begin by establishing that $\eta^* > 0$. Suppose to the contrary that $\eta^* = 0$, i.e. $\eta_k \xrightarrow{k \in \mathcal{K}} 0$. Then, in view of *Step 3(iv)*, without loss of generality, either

(i) $\widehat{d_k^0} \xrightarrow{k \in \mathcal{K}} 0$, with $\|\widehat{d_k^0}\| \leq \bar{D}$ and $\widehat{\lambda_k^0} \geq 0$, for all $k \in \mathcal{K}$, or

(ii) $\hat{d}_{k-1} \xrightarrow{k \in \mathcal{K}} 0$.

We first consider case (i). Let $\mathcal{K}' \subseteq \mathcal{K}$ be an infinite index set and $\Xi^0 \subseteq \Xi$ be such that $\hat{\Xi}_{k-1} \equiv \Xi^0$ for all $k \in \Xi^0$. Since $\widehat{d_k^0} \xrightarrow{k\in\mathcal{K}'} 0$, and since by construction $\widehat{d_k^0} = \widehat{d^0}(x_k, H_k, \Xi^0)$, it follows from the constraints of $LS^0(x_k, H_k, \Xi^0)$ that $\Xi^0 \subseteq \Xi_{\text{act}}(x^*)$. Thus, as a consequence of Assumption 2, $\{\ \nabla_x \phi(x^*, \xi) \mid \xi \in \Xi^0\ \}$ is a linearly independent set. Using an argument along the lines of that used in Lemma 47, we can show that $\{\widehat{\lambda_k^0}\}_{k\in\mathcal{K}'}$ is bounded, thus we assume without loss of generality that $\widehat{\lambda_k^0} \xrightarrow{k\in\mathcal{K}'} \widehat{\lambda^0}_* \geq 0$. Taking limits in the optimality conditions for $LS^0(x_k, H_k, \Xi^0)$ shows that x^* is a KKT point for (DSI) with multipliers

$$\lambda^{*,\xi} = \begin{cases} \widehat{\lambda^0}_*, & \xi \in \Xi^0, \\ 0, & \text{otherwise}, \end{cases}$$

a contradiction. Now consider case (ii). As $\hat{d}_{k-1} \xrightarrow{k\in\mathcal{K}} 0$, it follows that $x_{k-1} \xrightarrow{k\in\mathcal{K}} x^*$. In view of Lemma 49, x^* is a KKT point, which is again a contradiction. This establishes $\eta^* > 0$.

As $\eta^* > 0$, there exists $\underline{\eta} > 0$ such that $\eta_k \geq \underline{\eta}$ for all $k \in \mathcal{K}$. Now, since x^* is not KKT, in view of Lemma 49(iii), $\{\hat{d}_k\}_{k\in\mathcal{K}}$ is bounded away from zero, which implies $\{\hat{\gamma}_k\}_{k\in\mathcal{K}}$ is bounded away from zero. Thus, there exists $\bar{\gamma} < 0$ such that

$$\phi(x_k, \xi) + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \leq \bar{\gamma}\underline{\eta} < 0, \quad \forall \xi \in \Xi',$$

for all $k \in \mathcal{K}$. Therefore, there exists $\delta > 0$ and $\underline{k}$ such that for all $k \in \mathcal{K}$, $k > \underline{k}$,

$$\begin{aligned} \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \leq -\delta, \quad \forall \xi \in \Xi' \cap \Xi_{\text{act}}(x^*) \\ \phi(x_k, \xi) \leq -\delta, \quad \forall \xi \in \Xi' \setminus (\Xi' \cap \Xi_{\text{act}}(x^*)). \end{aligned} \tag{5.12}$$

Now let $\mathcal{Q} \triangleq \{x_k |\ k \in \mathcal{K}\} \cup \{x^*\}$, $\mathcal{D} \triangleq \{\hat{d}_k |\ k \in \mathcal{K}\} \cup \{\hat{d}^*\}$ and define

$$M(t, \xi) \triangleq \max_{x\in\mathcal{Q}}\ \max_{d\in\mathcal{D}}\ \max_{\zeta\in[0,1]} \|\nabla_x \phi(x + t\zeta d, \xi) - \nabla_x \phi(x, \xi)\| \cdot \|d\|,$$

which is well-defined and continuous in t for all $\xi \in \Xi'$, since $\mathcal{Q}$ and $\mathcal{D}$ are compact. Now for all $k \in \mathcal{K}$, $\xi \in \Xi'$ we have

$$\begin{aligned}
&\phi(x_k + t\hat{d}_k, \xi) - \phi(x_k, \xi) \\
&\quad = \int_0^1 \langle \nabla_x \phi(x_k + t\zeta \hat{d}_k, \xi), \hat{d}_k \rangle d\zeta \\
&\quad = t \left\{ \int_0^1 \langle \nabla_x \phi(x_k + t\zeta \hat{d}_k, \xi) - \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle d\zeta + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \right\} \\
&\quad \leq t \left\{ \sup_{\zeta \in [0,1]} \| \nabla_x \phi(x_k + t\zeta \hat{d}_k, \xi) - \nabla_x \phi(x_k, \xi) \| \cdot \| \hat{d}_k \| + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \right\} \\
&\quad \leq t \left\{ M(t, \xi) + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle \right\}. \qquad (5.13)
\end{aligned}$$

Further note that $M(0, \xi) = 0$, for all $\xi \in \Xi'$. For $\xi \in \Xi' \cap \Xi_{\text{act}}(x^*)$, define $\underline{t}_\xi$ such that $M(t, \xi) < \delta$ for all $t \in [0, \underline{t}_\xi]$. For all $\xi \in \Xi' \setminus (\Xi' \cap \Xi_{\text{act}}(x^*))$, our regularity assumptions and boundedness of $\{x_k\}$ and $\{\hat{d}_k\}$ imply there exist $M_{1,\xi} > 0$ and $M_{2,\xi} > 0$ such that

$$|\langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle| \leq M_{1,\xi}, \ \forall k, \quad \text{and} \quad \max_{t \in [0,1]} |M(t, \xi)| \leq M_{2,\xi}.$$

For such ξ, define $\underline{t}_\xi = \delta / (M_{1,\xi} + M_{2,\xi})$. Then $t\{M(t, \xi) + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k \rangle\} \leq \delta$, for all $t \in [0, \underline{t}_\xi]$, $\xi \in \Xi' \setminus (\Xi' \cap \Xi_{\text{act}}(x^*))$. Finally, set $\underline{t} = \max_{\xi \in \Xi'} \underline{t}_\xi$. From (5.13) and (5.12) it is easily verified that $\underline{t}$ is as claimed. □

Lemma 52. $\liminf_k \hat{v}_k = 0$.

Proof. We argue by contradiction. That is, suppose

$$\liminf_k \hat{v}_k = \hat{v}^* > 0. \qquad (5.14)$$

As all sequences of interest are bounded, there exists an infinite index set $\mathcal{K}$ such that

$$\begin{aligned} \hat{v}_k &\xrightarrow{k\in\mathcal{K}} \hat{v}^*, & x_k &\xrightarrow{k\in\mathcal{K}} x^* \in X, & H_k &\xrightarrow{k\in\mathcal{K}} H^* \in \mathcal{H}, \\ \hat{v}_{k+1} &\xrightarrow{k\in\mathcal{K}} \hat{v}^*_+, & \hat{d}_k &\xrightarrow{k\in\mathcal{K}} \hat{d}^*, & \hat{d}_{k+1} &\xrightarrow{k\in\mathcal{K}} \hat{d}^*_+, \\ \hat{\gamma}_k &\xrightarrow{k\in\mathcal{K}} \hat{\gamma}^*, & \hat{\gamma}_{k+1} &\xrightarrow{k\in\mathcal{K}} \hat{\gamma}^*_+, & \eta_k &\xrightarrow{k\in\mathcal{K}} \eta^* \geq 0, \end{aligned}$$

and $\hat{\Xi}^b_k \equiv \Xi'$, for all $k \in \mathcal{K}$. Since $\hat{\Xi}^b_k$ consists of the indices of the binding constraints for $\widehat{QP}(x_k, H_k, \eta_k, \Xi_k)$, $(\hat{d}_k, \hat{\gamma}_k)$ solves $\widehat{QP}(x_k, H_k, \eta_k, \Xi')$, for all $k \in \mathcal{K}$, and we may assume without loss of generality that $\Xi_k \equiv \Xi'$ for all $k \in \mathcal{K}$. In view of Lemma 47, $(\hat{d}^*, \hat{\gamma}^*)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*, \Xi')$, and by Lemmas 49 and 43, $\hat{\gamma}^* < 0$. Thus, the objective descent constraint in $\widehat{QP}(x^*, H^*, \eta^*, \Xi')$ gives

$$\langle \nabla f(x^*), \hat{d}^* \rangle < 0.$$

Now, in view of Lemmas 48(*iii*) and 49, $t_k \xrightarrow{k\in\mathcal{K}} 0$. Without loss of generality, we assume that $t_k < \min\{\delta, \underline{t}\}$, for all $k \in \mathcal{K}$, where δ is as defined in the algorithm statement and $\underline{t}$ is as given by Lemma 51. Note that since $t_k < \delta < 1$, at least one of the line search conditions of *Step 2* is not satisfied at $\bar{x}_{k+1} = x_k + \frac{t_k}{\beta}\hat{d}_k$ for all $k \in \mathcal{K}$. As $\alpha < 1$ a standard argument may be used to show that condition (5.4) is violated at $\bar{x}_{k+1}$ only finitely many times. Thus we assume that condition (5.5) causes the line search failure for all $k \in \mathcal{K}$, i.e.

$$\phi(\bar{x}_{k+1}, \bar{\xi}_k) > 0, \quad \forall k \in \mathcal{K}.$$

As there are only finitely many constraints, we may assume without loss of generality that $\bar{\xi}_k \equiv \bar{\xi}$, for all $k \in \mathcal{K}$. In view of Lemma 51, $\bar{\xi} \notin \Xi'$. As a consequence, by *Step 3(iv)*, $H_{k+1} = H_k$ and $\eta_{k+1} = \eta_k$ for all $k \in \mathcal{K}$. Further, we assume without loss of generality that

$$\Xi_{k+1} \equiv \Xi'' \supseteq \Xi' \cup \{\bar{\xi}\}, \quad \forall k \in \mathcal{K}.$$

It follows that $(\hat{d}_{k+1}, \hat{\gamma}_{k+1})$ solves $\widehat{QP}(x_{k+1}, H_k, \eta_k, \Xi'')$, for all $k \in \mathcal{K}$.

Note that in view of Lemma 48(*iii*), $x_{k+1} \overset{k\in\mathcal{K}}{\longrightarrow} x^*$, hence by Lemma 47, $(\hat{d}^*_+, \hat{\gamma}^*_+)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*, \Xi'')$. Since $\phi(\bar{x}_{k+1}, \bar{\xi}) > 0$ and $\phi(x_{k+1}, \bar{\xi}) \leq 0$, for all $k \in \mathcal{K}$, it follows that $\phi(x^*, \bar{\xi}) = 0$. Expanding $\phi(\bar{x}_{k+1}, \bar{\xi}) - \phi(x_{k+1}, \bar{\xi})$ and taking limits shows that

$$\langle \nabla_x \phi(x^*, \bar{\xi}), \hat{d}^* \rangle \geq 0.$$

It follows that, since $\hat{\gamma}^* < 0$ and $\eta^* > 0$ (by Lemma 51), $(\hat{d}^*, \hat{\gamma}^*)$ is infeasible for $\widehat{QP}(x^*, H^*, \eta^*, \Xi'')$. Finally, in view of Lemma 50, $\hat{v}^*_+ < \hat{v}^*$. As this contradicts (5.14), the proof is complete. □

Corollary 1. *There exists an accumulation point of* $\{x_k\}$ *which is a KKT point for* (DSI)*.*

Proof. Follows immediately from Lemmas 49 and 52. □

Define the Lagrangian function for (DSI) as

$$L(x, \lambda) \stackrel{\Delta}{=} f(x) + \sum_{\xi \in \Xi} \lambda^\xi \phi(x, \xi).$$

In order to show that the entire sequence converges to a KKT point x^*, we strengthen our assumptions as follows.

Assumption 1′: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $\phi(\cdot, \xi) : \mathbb{R}^n \to \mathbb{R}$, $\xi \in \Xi$ are twice continuously differentiable.

Assumption 5: Some accumulation point x^* of $\{x_k\}$ which is a KKT point for (DSI) also satisfies the second order sufficiency conditions with strict complementary slackness, i.e. there exists $\lambda^* \in \mathbb{R}^{|\Xi|}$ satisfying (5.1) as well as

- $\nabla^2_{xx} L(x^*, \lambda^*)$ is positive definite on the subspace

$$\{h \mid \langle \nabla_x \phi(x^*, \xi), h \rangle = 0,\ \forall \xi \in \Xi_{\text{act}}(x^*)\},$$

- and $\lambda^{*,\xi} > 0$ for all $\xi \in \Xi_{\text{act}}(x^*)$.

It is well-known that such an assumption implies that x^* is an isolated KKT point for (DSI) as well as an isolated local minimizer. The following theorem is the main result of this section.

Theorem 7. *The sequence $\{x_k\}$ generated by algorithm* **FSQP′-MC** *converges to a strict local minimizer x^* of (DSI).*

Proof. First we show that there exists a neighborhood of x^* in which no other accumulation points of $\{x_k\}$ can exist, KKT points or not. As x^* is a strict local minimizer, there exists $\epsilon > 0$ such that $f(x) > f(x^*)$ for all $x \neq x^*$, $x \in \mathcal{S} \overset{\Delta}{=} B(x^*, \epsilon) \cap X$, where $B(x^*, \epsilon)$ is the open ball of radius ϵ centered at x^*. Proceeding by contradiction, suppose $x' \in B(x^*, \epsilon)$, $x' \neq x^*$, is another accumulation point of $\{x_k\}$. Feasibility of the iterates implies that $x' \in \mathcal{S}$. Thus $f(x') > f(x^*)$, which is in contradiction with Lemma 48(*ii*). Next, in view of Lemma 48(*iii*), $(x_{k+1} - x_k) \to 0$. Suppose $\mathcal{K}$ is an infinite index set such that $x_k \xrightarrow{k \in \mathcal{K}} x^*$. Then there exists k_1 such that $\|x_k - x^*\| < \epsilon/4$, for all $k \in \mathcal{K}$, $k \geq k_1$. Further, there exists k_2 such that $\|x_{k+1} - x_k\| < \epsilon/4$, for all $k > k_2$. Therefore, if there were another accumulation point outside of $B(x^*, \epsilon)$, then the sequence would have to pass through the compact set $\overline{B(x^*, \epsilon)} \setminus B(x^*, \epsilon/4)$ infinitely many times. This contradicts the established fact that there are no accumulation points of $\{x_k\}$, other than x^*, in $B(x^*, \epsilon)$. □

5.3.2 Local Convergence

We have thus shown that, with a likely dramatically reduced amount of work per iteration, global convergence can be preserved. This would be of little interest, though, if the speed of convergence were to suffer significantly. In this section we establish that, under a few additional assumptions, the sequence $\{x_k\}$ generated by a slightly modified version of algorithm **FSQP′-MC** (to avoid the Maratos effect) exhibits 2-step superlinear convergence. To do this, the bulk of our effort is focused on showing that for k large the set of constraints $\hat{\Xi}_k^b$ which affect the search direction is precisely the set of active constraints at the solution, i.e. $\Xi_{\text{act}}(x^*)$. In addition, we show that, eventually, no constraints outside of $\Xi_{\text{act}}(x^*)$ affect the line search, and that H_k is updated normally at every iteration. Thus, for k large enough, the algorithm behaves as if it were solving the problem

$$\begin{aligned} &\min \quad && f(x) \\ &\text{s.t.} \quad && \phi(x,\xi) \le 0, \qquad \xi \in \Xi_{\text{act}}(x^*), \end{aligned} \tag{P^*}$$

using *all* constraints at every iteration. Establishing this allows us to apply the results of Section 3.3.2 concerning local convergence rates.

Lemma 53. *Suppose $\mathcal{K}$ is an infinite index set such that $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \ge 0$, $H_k \xrightarrow{k\in\mathcal{K}} H^* \in \mathcal{H}$, and $\Xi_k \equiv \Xi^*$, for all $k \in \mathcal{K}$. Then $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$ satisfies the strong second order sufficiency conditions with strict complementary slackness, the gradients of the active constraints are linearly independent, and $\Xi_{act}(x^*) \subseteq \Xi^*$.*

Proof. Consider the Lagrangian function for $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$

$$\begin{aligned}\hat{L}^*(\hat{d}, \hat{\gamma}, \hat{\mu}, \hat{\lambda}) \triangleq\ & \frac{1}{2}\langle \hat{d}, H^* \hat{d}\rangle + \hat{\gamma} \\ & + \hat{\mu}\left(\langle \nabla f(x^*), \hat{d}\rangle - \hat{\gamma}\right) + \sum_{\xi \in \Xi^*} \hat{\lambda}^\xi \left(\phi(x^*, \xi) + \langle \nabla_x \phi(x^*, \xi), \hat{d}\rangle - \hat{\gamma} \cdot \eta^*\right).\end{aligned}$$

In view of Lemmas 52 and 49, we may assume without loss of generality that $(\hat{d}_k, \hat{\gamma}_k) \xrightarrow{k \in \mathcal{K}} (0, 0)$. Thus, $(0, 0)$ is the unique solution of $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$ and it is not difficult to show that (plug $(0, 0)$ into the constraints of $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$) the set of active constraint gradients for $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$ is

$$N^* \triangleq \left\{ \begin{pmatrix} \nabla f(x^*) \\ 1 \end{pmatrix}, \begin{pmatrix} \nabla_x \phi(x^*, \xi) \\ -\eta^* \end{pmatrix}, \xi \in \Xi^* \cap \Xi_{\text{act}}(x^*) \right\}.$$

Using an argument identical to that given in the proof of Lemma 11 in Chapter 3, we can show that N^* is a linearly independent set. Further, the argument from Lemma 12 in Chapter 3 may be used to show that the Hessian of the Lagrangian $\nabla^2 \hat{L}^*(0, 0, \hat{\mu}^*, \hat{\lambda}^*)$ is positive definite on $N^{*\perp}$. Thus, the second order sufficiency conditions hold.

Let $\hat{\mu}^*$, $\hat{\lambda}^{*,\xi}$, $\xi \in \Xi^*$, denote the unique (since N^* is linearly independent) multipliers from $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$. We now show that strict complementary slackness holds and $\Xi_{\text{act}}(x^*) \subseteq \Xi^*$. An identical argument to that used in Lemma 12 in Chapter 3 can be used to show that $\hat{\mu}^* > 0$. In view of Assumption 2, the multipliers $\lambda^{*,\xi}$, $\xi \in \Xi$, are unique for (DSI) at x^*. It thus follows from the optimality conditions (5.2) that

$$\lambda^{*,\xi} = \begin{cases} \dfrac{\hat{\lambda}^{*,\xi}}{\hat{\mu}^*}, & \xi \in \Xi^*, \\ 0, & \text{otherwise}. \end{cases}$$

Finally, as a consequence of Assumption 5, $\Xi_{\text{act}}(x^*) \subseteq \Xi^*$ and

$$\frac{\hat{\lambda}^{*,\xi}}{\hat{\mu}^*} > 0, \quad \forall \xi \in \Xi_{\text{act}}(x^*).$$

Therefore, strict complementary slackness holds for $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$. It also follows that $\Xi_{\text{act}}(x^*) \subseteq \Xi^*$. □

Lemma 54. *There exists an infinite index set $\mathcal{K}$ such that $\Xi_{act}(x^*) \subseteq \hat{\Xi}^b_k$ for all $k \in \mathcal{K}$.*

Proof. Let $\mathcal{K}$ be as in the previous Lemma. We may apply the classical result of Robinson [68] to conclude

$$\hat{\mu}_k \xrightarrow{k \in \mathcal{K}} \hat{\mu}^*,$$
$$\hat{\lambda}^{\xi}_k \xrightarrow{k \in \mathcal{K}} \hat{\lambda}^{*,\xi}, \quad \forall \xi \in \Xi^*,$$

where $\hat{\mu}_k$, $\hat{\lambda}^{\xi}_k$, $\xi \in \Xi^*$ are the QP multipliers from $\widehat{QP}(x_k, H_k, \eta_k, \Xi^*)$ and $\hat{\mu}^* > 0$, $\hat{\lambda}^{*,\xi}$, $\xi \in \Xi^*$ are the QP multipliers from $\widehat{QP}(x^*, H^*, \eta^*, \Xi^*)$. Note further that uniqueness of the multipliers $\lambda^{*,\xi}$, $\xi \in \Xi$, for (DSI) at x^* and the optimality conditions (5.2) give

$$\lambda^{*,\xi} = \begin{cases} \dfrac{\hat{\lambda}^{*,\xi}}{\hat{\mu}^*}, & \xi \in \Xi^*, \\ 0, & \text{otherwise.} \end{cases}$$

As $\hat{\mu}^* > 0$, strict complementary slackness for (DSI) (Assumption 5) implies $\hat{\lambda}^{*,\xi} > 0$, for all $\xi \in \Xi_{\text{act}}(x^*)$. Therefore, for all $k \in \mathcal{K}$, k sufficiently large, $\Xi_{\text{act}}(x^*) \subseteq \hat{\Xi}^b_k$. □

Before stating and proving the next lemma, we note that as a consequence of Lemma 48(i), there exists $\bar{\eta} > 0$ such that

$$\eta_k \leq \bar{\eta},$$

for all k.

Lemma 55. *Given $\epsilon > 0$, there exists $\delta > 0$ such that for every $x \in X$ satisfying $\|x - x^*\| < \delta$, every $\eta \in [0, \bar{\eta}]$, every $H \in \mathcal{H}$, and every $\Xi' \subseteq \Xi$ with $\Xi_{act}(x^*) \subseteq \Xi'$, all $\xi \in \Xi_{act}(x^*)$ are binding for $\widehat{QP}(x, H, \eta, \Xi')$ and $\|\hat{d}(x, H, \eta, \Xi')\| < \epsilon$.*

Proof. Given $H \in \mathcal{H}$, $\eta \in [0, \bar{\eta}]$, and $\Xi' \subseteq \Xi$ such that $\Xi_{\text{act}}(x^*) \subseteq \Xi'$, Lemma 44 implies that $\hat{d}(x^*, H, \eta, \Xi') = 0$. Since $\mathcal{H}$ and $[0, \bar{\eta}]$ are compact, Lemma 53 and Assumption 5 allows us to apply Theorem 2.1 of [68] to conclude that, given $\epsilon > 0$, there exists $\delta_{\Xi'} > 0$ such that for all x satisfying $\|x - x^*\| < \delta_{\Xi'}$ and all $H \in \mathcal{H}$, $\eta \in [0, \bar{\eta}]$, the QP multipliers from $\widehat{QP}(x, H, \eta, \Xi')$ are positive for all $\xi \in \Xi_{\text{act}}(x^*)$ and $\|\hat{d}(x, H, \eta, \Xi')\| < \epsilon$. As Ξ is a finite set, δ may be chosen independent of Ξ'. □

Lemma 56. *For k sufficiently large $\Xi_{act}(x^*) \subseteq \hat{\Xi}_k^b$.*

Proof. For an arbitrary $\epsilon > 0$, let $\delta > 0$ be as given by Lemma 55. In view of Theorem 7, there exists $\underline{k}$ such that $\|x_k - x^*\| < \delta$ for all $k \geq \underline{k}$. By Lemma 54, there exists an infinite index set $\mathcal{K}$ such that $\Xi_{\text{act}}(x^*) \subseteq \hat{\Xi}_k^b$, for all $k \in \mathcal{K}$. Choose $\underline{k}' \geq \underline{k}$, $\underline{k}' \in \mathcal{K}$. It follows that $\Xi_{\text{act}}(x^*) \subseteq \Xi_{\underline{k}'+1}$. The result follows by induction and Lemma 55. □

Lemma 57. $\hat{d}_k \longrightarrow 0$.

Proof. Follows immediately from Lemma 56, *Step 3(iii)* of algorithm **FSQP′-MC**, Assumption 4, and Lemma 55. □

Lemma 58. *For k large enough,*

(i) $\hat{\Xi}_k^b = \Xi_{act}(x^*)$, *and*

(ii) $\phi(x_k + t\hat{d}_k, \xi) \leq 0$ *for all* $t \in [0, 1]$, $\xi \in \Xi \setminus \Xi_{act}(x^*)$.

Proof. For (i), in view of Lemma 56, it suffices to show that, for k sufficiently large, $\hat{\Xi}_k^b \subseteq \Xi_{\text{act}}(x^*)$. Suppose $\xi' \in \Xi \setminus \Xi_{\text{act}}(x^*)$, i.e. $\phi(x^*, \xi') < 0$. Since

$x_k \longrightarrow x^*$, by continuity we have $\phi(x_k, \xi') < 0$ for all k sufficiently large. In view of Lemma 57, for k sufficiently large we have

$$\phi(x_k, \xi') + \langle \nabla_x \phi(x_k, \xi'), \hat{d}_k \rangle < 0.$$

Therefore, $\hat{\lambda}_k^{\xi'} = 0$ (hence $\xi' \notin \hat{\Xi}_k^b$) for all k sufficiently large. Part (*ii*) follows from Theorem 7, Lemma 57, and our regularity assumptions. $\square$

In order to achieve superlinear convergence, it is crucial that a unit step, i.e. $t_k = 1$, always be accepted for all k sufficiently large. Again, we will include a second order correction such as that used in Chapter 3. Specifically, at iteration k, let $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{\Xi}_k)$ be the solution of $\widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{\Xi}_k)$, defined for $\tau \in (2, 3)$ as follows

$$\begin{array}{rl} \min & \frac{1}{2}\langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d}\,)\rangle + \langle \nabla f(x_k), \hat{d}_k + \tilde{d}\,\rangle \\ \text{s.t.} & \phi(x_k + \hat{d}_k, \xi) + \langle \nabla_x \phi(x_k, \xi), \hat{d}_k + \tilde{d}\rangle = -\|\hat{d}_k\|^\tau, \quad \forall \xi \in \hat{\Xi}_k, \\ & \hfill \widetilde{LS}(x_k, \hat{d}_k, H_k, \hat{\Xi}_k) \end{array}$$

if it exists and has norm less that $\min\{\|\hat{d}_k\|, C\}$, where C is a large number. Otherwise, set $\tilde{d}_k = 0$. The following step is added to algorithm **FSQP′-MC**:

Step 1(iii). **compute** $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{\Xi}_k)$.

In addition, the line search criterion (5.4) and (5.5) are replaced with

$$f(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq f(x_k) + \alpha t \langle \nabla f(x_k), \hat{d}_k \rangle, \tag{5.15}$$

and

$$\phi(x_k + t\hat{d}_k + t^2\tilde{d}_k) \leq 0, \quad \forall \xi \in \Xi. \tag{5.16}$$

Finally, the condition (5.6) is replaced with

$$\phi\left(x_k + \frac{t_k}{\beta}\hat{d}_k + \left(\frac{t_k}{\beta}\right)^2 \tilde{d}_k, \bar{\xi}\right) > 0. \tag{5.17}$$

With some effort, it can be shown that these modifications do not affect any of the results obtained to this point. Hence, it is established that for k large enough, the modified algorithm **FSQP′-MC** behaves identically to that given in Chapter 3, applied to (P^*).

Assumption 1 is now further strengthened and a new assumption concerning the Hessian approximations H_k is given. These assumptions allow us to use the local convergence rate result from Chapter 3.

Assumption 1″: The functions $f : \mathbb{R}^n \to \mathbb{R}$, and $\phi(\cdot, \xi) : \mathbb{R}^n \to \mathbb{R}$, $\xi \in \Xi$, are three times continuously differentiable.

Assumption 6: As a result of the update rule chosen for *Step 3(iv)*, H_k approaches the Hessian of the Lagrangian in the sense that

$$\lim_{k\to\infty} \frac{\|P_k(H_k - \nabla^2_{xx} L(x^*, \lambda^*))P_k\hat{d}_k\|}{\|\hat{d}_k\|} = 0,$$

where λ^* is the KKT multiplier vector associated with x^* and

$$P_k \triangleq I - R_k(R_k^T R_k)^{-1}R_k^T$$

with $R_k = [\nabla_x \phi(x_k, \xi) \mid \xi \in \Xi_{\text{act}}(x^*)]$.

Theorem 8. *For all k sufficiently large, the unit step $t_k = 1$ is accepted in Step 2. Further, the sequence $\{x_k\}$ converges to x^* 2-step superlinearly, i.e.*

$$\lim_{k\to\infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

5.4 Extensions to Constrained Mini-Max

The algorithm we have discussed may be extended following the scheme of [78] to handle problems with many objective functions, i.e. large-scale constrained mini-max. Specifically, consider the problem

$$\begin{aligned} \min \quad & \max_{\omega \in \Omega} f(x,\omega) \\ \text{s.t.} \quad & \phi(x,\xi) \leq 0, \quad \forall \xi \in \Xi, \end{aligned}$$

where Ω and Ξ are finite (again, presumably large) sets, and $f : \mathbb{R}^n \times \Omega \to \mathbb{R}$ and $\phi : \mathbb{R}^n \times \Xi \to \mathbb{R}$ are both three times continuously differentiable with respect to their first argument. Given $\Omega' \subseteq \Omega$, define

$$F_{\Omega'}(x) \stackrel{\Delta}{=} \max_{\omega \in \Omega'} f(x,\omega).$$

Given a direction $d \in \mathbb{R}^n$, define a first-order approximation of $F_{\Omega'}(x+d) - F_{\Omega'}(x)$ by

$$F'_{\Omega'}(x,d) \stackrel{\Delta}{=} \max_{\omega \in \Omega'} \{ f(x+d,\omega) + \langle \nabla_x f(x,\omega), d \rangle \} - F_{\Omega'}(x).$$

Define $(\hat{d}(x,H,\eta,\Omega',\Xi'), \hat{\gamma}(x,H,\eta,\Omega',\Xi')) \in \mathbb{R}^n \times \mathbb{R}$ as the solution of the QP

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle \hat{d}, H\hat{d} \rangle + \hat{\gamma} \\ \text{s.t.} \quad & F'_{\Omega'}(x,\hat{d}) \leq \hat{\gamma}, \qquad\qquad\qquad\qquad \widehat{QP}(x,H,\eta,\Omega',\Xi') \\ & \phi(x,\xi) + \langle \nabla_x \phi(x,\xi), \hat{d} \rangle \leq \hat{\gamma} \cdot \eta, \quad \forall \xi \in \Xi'. \end{aligned}$$

The second order correction $\tilde{d}(x,\hat{d},H,\Omega',\Xi')$ is computed as the solution, if it exists, of the equality constrained QP

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle \hat{d}+\tilde{d}, H(\hat{d}+\tilde{d}) \rangle + \tilde{\gamma} \\ \text{s.t.} \quad & f(x+\hat{d},\omega) + \langle \nabla_x f(x,\omega), \tilde{d} \rangle = F_{\Omega'}(x) + \tilde{\gamma}, \quad \forall \omega \in \Omega' \\ & \phi(x+\hat{d},\xi) + \langle \nabla_x \phi(x,\xi), \tilde{d} \rangle = -\|\hat{d}\|^\tau, \qquad \forall \xi \in \Xi', \\ & \qquad\qquad\qquad\qquad\qquad\qquad \widetilde{LS}(x,\hat{d},H,\Omega',\Xi') \end{aligned}$$

where $\tau \in (2,3)$ and again, if the QP has no solution, or if the solution has norm greater than $\min\{\|\hat{d}\|, C\}$, we set $\tilde{d}(x,d,H,\Omega',\Xi') = 0$. Finally, the estimate of the SQP direction $\widehat{d^0}(x,H,\Omega',\Xi')$ is taken from the solution

$$(\widehat{d^0}, \widehat{\gamma^0}) = (\widehat{d^0}(x,H,\Omega',\Xi'), \widehat{\gamma^0}(x,H,\Omega',\Xi')) \in \mathbb{R}^n \times \mathbb{R},$$

if it exists, of the equality constrained QP

$$\begin{array}{lll} \min & \frac{1}{2}\langle\widehat{d^0}, H\widehat{d^0}\rangle + \widehat{\gamma^0} & \\ \text{s.t.} & f(x,\omega) + \langle\nabla_x f(x,\omega), \widehat{d^0}\rangle = F_{\Omega'}(x) + \widehat{\gamma^0}, \quad \forall\omega\in\Omega' & LS^0(x,H,\Omega',\Xi') \\ & \phi(x,\xi) + \langle\nabla_x \phi(x,\xi), \widehat{d^0}\rangle = 0, \qquad\qquad\qquad \forall\xi\in\Xi', & \end{array}$$

As was the case with the constraints, at iteration k, only a subset $\Omega_k \subseteq \Omega$ will be used to compute the search directions. In order to describe the update rules for Ω_k, following [78], we define a few index sets for the objectives (in direct analogy with the index sets for the constraints as introduced in Section 5.2). The set of indices of "maximizing" objectives is defined in the obvious manner as

$$\Omega_{\max}(x) \overset{\Delta}{=} \{\omega\in\Omega \mid f(x,\omega) = F_\Omega(x)\}.$$

At iteration k, let $\hat{\mu}_k^\omega$, $\omega\in\Omega_k$, be the multipliers from $\widehat{QP}(x_k, H_k, \eta_k, \Omega_k, \Xi_k)$ associated with the objective functions. The set of indices of objective functions which affected the computation of the search direction $\hat{d}_k$ is given by

$$\hat{\Omega}_k^b \overset{\Delta}{=} \{\omega\in\Omega_k \mid \hat{\mu}_k^\omega > 0\ \}.$$

The line search criterion (5.15) is replaced with

$$F_\Omega(x_k + t\hat{d}_k + t^2\tilde{d}_k) \le F_\Omega(x_k) + \alpha t F'_{\Omega_k}(x_k, \hat{d}_k). \tag{5.18}$$

If $t_k < 1$ and the truncation is due to an objective function, then define $\bar{\omega}\in\Omega$ as an index such that

$$f\left(x_k + \frac{t_k}{\beta}\hat{d}_k + \left(\frac{t_k}{\beta}\right)^2 \tilde{d}_k, \bar{\omega}\right) > F_\Omega(x_k) + \alpha\frac{t_k}{\beta}F'_{\Omega_k}(x_k, \hat{d}_k). \tag{5.19}$$

Remark: Note that we use $F'_{\Omega_k}(x_k, \hat{d}_k)$ in the line search descent criterion instead of $F'_\Omega(x_k, \hat{d}_k)$. This allows us to skip the evaluation of the objective function gradients $\nabla_x f(x_k, \omega)$, $\omega \in \Xi \setminus \Xi_k$, potentially saving a great deal of effort in the case that gradient evaluations are expensive.

Finally, define

$$\hat{\Omega}_k \triangleq \{ \ \omega \in \Omega_k \mid f(x_k, \omega) + \langle \nabla_x f(x_k, \omega), \hat{d}_k \rangle = F_{\Omega_k}(x_k) + \hat{\gamma}_k \ \}.$$

We are now in a position to state the extended algorithm.

Algorithm FSQP′-MOC

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $0 < \delta \ll 1$, $\epsilon_\ell > 0$, $0 < \underline{C} \leq \overline{C}$, $\bar{D} > 0$.

Data. $x_0 \in X$, $0 < H_0 = H_0^T \in \mathbb{R}^{n \times n}$.

Step 0 - Initialization. **set** $k \leftarrow 0$ and **choose** $\Omega_0 \supseteq \Omega_{\max}(x_0)$, $\Xi_0 \supseteq \Xi_{\text{act}}(x_0)$.

Step 1 - Computation of search directions.

(i) **compute** $\hat{d}_k = \hat{d}(x_k, H_k, \eta_k, \Omega_k, \Xi_k)$.

(ii) **if** $\hat{d}_k = 0$, **then stop**.

(iii) **compute** $\tilde{d}_k = \tilde{d}(x_k, \hat{d}_k, H_k, \hat{\Omega}_k, \hat{\Xi}_k)$.

Step 2 - Line search. **compute** t_k, the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying (5.18) and (5.16).

Step 3 - Updates.

(i). **set** $x_{k+1} \leftarrow x_k + t_k \hat{d}_k$.

(ii). **if** $t_k < 1$ and (5.18) was violated at $\bar{x}_{k+1} = x_k + \frac{t_k}{\beta}\hat{d}_k + \left(\frac{t_k}{\beta}\right)^2 \tilde{d}_k$, **then** let $\bar{\omega}$ be such that (5.19) holds.

if (5.16) was violated at $\bar{x}_{k+1}$, **then** let $\bar{\xi}$ be such that (5.17) holds.

(iii). **pick**

$$\Omega_{k+1} \supseteq \Omega_{\max}(x_{k+1}) \cup \Omega_k^b, \quad \text{and}$$

$$\Xi_{k+1} \supseteq \Xi_{\text{act}}(x_{k+1}) \cup \hat{\Xi}_k^b.$$

if $t_k < 1$ and (5.19) holds for some $\bar{\omega} \in \Omega$, **then set** $\Omega_{k+1} \leftarrow \Omega_{k+1} \cup \{\bar{\omega}\}$. **if** $t_k < 1$ and (5.17) holds for some $\bar{\xi} \in \Xi$, **then set** $\Xi_{k+1} \leftarrow \Xi_{k+1} \cup \{\bar{\xi}\}$.

(iv). **if** $t_k \leq \delta$ and $\bar{\omega} \notin \Omega_k$ or $\bar{\xi} \notin \Xi_k$, **then set** $H_{k+1} \leftarrow H_k$, $\eta_{k+1} \leftarrow \eta_k$. **else,**

(a) **compute** a new symmetric positive definite estimate H_{k+1} of the Hessian of the Lagrangian.

(b) **select** $C_{k+1} \in [\underline{C}, \overline{C}]$.

- $*$ **if** $(\|\hat{d}_k\| < \epsilon_\ell)$ **then**
    - $\cdot$ **compute**, if possible,[4] $\widehat{d^0_{k+1}} = \widehat{d^0}(x_{k+1}, H_{k+1}, \hat{\Omega}_k, \hat{\Xi}_k)$, and the associated multipliers $\widehat{\mu^0_{k+1}} \in \mathbb{R}^{|\hat{\Omega}_k|}$ and $\widehat{\lambda^0_{k+1}} \in \mathbb{R}^{|\hat{\Xi}_k|}$.
    - $\cdot$ **if** $\Big(\widehat{d^0_{k+1}}$ exists and $\|\widehat{d^0_{k+1}}\| \leq \bar{D}$ and $\widehat{\lambda^0_{k+1}} \geq 0$ and $\widehat{\mu^0_{k+1}} \geq 0$ and $F_\Omega(x_{k+1}) = F_{\hat{\Omega}_k}(x_{k+1})\Big)$ **then set**

    $$\eta_{k+1} \leftarrow C_{k+1} \cdot \|\widehat{d^0_{k+1}}\|^2.$$

    - $\cdot$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \|\hat{d}_k\|^2$.
- $*$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \epsilon_\ell^2$.

(v). **set** $k \leftarrow k + 1$ and **goto** *Step 1.*

[4] That is, if $LS^0(x_{k+1}, H_{k+1}, \hat{\Omega}_k, \hat{\Xi}_k)$ is non-degenerate.

5.5 Implementation and Numerical Results

We only discuss the implementation details for **FSQP′-MC** here. The details for **FSQP′-MOC** are similar. The implementation allows for multiple discretized SIP constraints and contains special provisions for those which are affine in x. Specifically, problem (DSI) is generalized to

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \phi_j(x,\xi) \overset{\Delta}{=} \langle c_j(\xi), x\rangle - d_j(\xi) \le 0, \quad && \forall \xi \in \Xi^{(j)},\ j = 1,\ldots,m_\ell, \\ & \phi_j(x,\xi) \le 0, && \forall \xi \in \Xi^{(j)},\ j = m_\ell + 1,\ldots,m, \end{aligned}$$

where $c_j : \Xi_\ell^{(j)} \longrightarrow \mathbb{R}^n$, $j = 1,\ldots,m_\ell$, $d_j : \Xi_\ell^{(j)} \longrightarrow \mathbb{R}$, $j = 1,\ldots,m_\ell$, and $\Xi^{(j)}$ is finite, $j = 1,\ldots,m$. The assumptions and algorithm statement are generalized in the obvious manner. Analogous to what was done in Chapter 3, no tilting is required for the affine constraints. As far as the analysis of Section 5.3 is concerned, such a formulation could readily be adapted to the format of (DSI) by grouping all constraints together, i.e. letting $\Xi = \cup_{j=1}^m \Xi^{(j)}$. The arguments would have to be modified slightly to account for the fact that no tilting is done for the affine constraints. Since they are incorporated directly into the sub-problems, though, it should be obvious that tilting is not necessary.

Recall that it is only required that Ξ_k *contain* certain subsets of Ξ. The algorithm allows for additional elements of Ξ to be included in order to speed up initial convergence. Of course, there is a trade-off between speeding up initial convergence and increasing (i) the number of gradient evaluations and (ii) the size of the QPs. In the implementation, heuristics are applied to add potentially useful elements to Ξ_k (see, e.g. , [71] for a discussion of such heuristics). In the case of discretized SIP, one may wish to exploit the knowledge that adjacent discretization points are likely to be closely related. Following [78, 49, 19], for some $\epsilon > 0$, the implementation includes in Ξ_k the set $\Xi_\epsilon^{\ell\ell m}(x_k)$ of ϵ-active "left local maximizers" at x_k. At a point $x \in X$, for $j = 1, \ldots, m$, define the ϵ-active discretization points as

$$\Xi_\epsilon^{(j)}(x) \triangleq \{\xi \in \Xi^{(j)} \mid \phi_j(x, \xi) \geq -\epsilon\}.$$

Such a discretization point $\xi_i^{(j)} \in \Xi^{(j)} = \{\xi_1^{(j)}, \ldots, \xi_{|\Xi^{(j)}|}^{(j)}\}$ is a left local maximizer if it satisfies *one* of the following three conditions: (i) $i \in \{2, \ldots, |\Xi^{(j)}| - 1\}$ and

$$\phi_j(x, \xi_i^{(j)}) > \phi_j(x, \xi_{i-1}^{(j)}) \tag{5.20}$$

and

$$\phi_j(x, \xi_i^{(j)}) \geq \phi_j(x, \xi_{i+1}^{(j)}); \tag{5.21}$$

(ii) $i = 1$ and (5.21); (iii) $i = |\Xi^{(j)}|$ and (5.20). The set $\Xi_\epsilon^{\ell\ell m}(x)$ is the set of all left local maximizers in $\Xi_\epsilon(x) = \cup_{j=1}^m \Xi_\epsilon^{(j)}(x)$. The first part of the update (i.e. before updates due to line search violations) in *Step 3*(iii) of the algorithm becomes

$$\Xi_{k+1} = \Xi_{\text{act}}(x_k) \cup \Xi_k^b \cup \Xi_\epsilon^{\ell\ell m}(x_k).$$

Finally, we have found that in practice, including the end-points (whether or not they are close to being active) during the first iteration often leads to a better initial search direction. Thus we set

$$\Xi_0 = \Xi_{\text{act}}(x_0) \cup \Xi_\epsilon^{\ell\ell m}(x_0) \cup \left(\bigcup_{j=1}^m (\{\xi_1^{(j)}\} \cup \{\xi_{|\Xi^{(j)}|}^{(j)}\}) \right).$$

The implementation handles simple bounds on the variables, updates on the coefficients C_k^j, and updates of the Hessian estimate H_k in a manner analogous to that in the basic algorithm **FSQP′** (see Section 3.4). Note that the implementation maintains a separate tilting parameter η_k^j, $j = 1, \ldots, m - m_\ell$, for each discretized nonlinear SIP constraint. As in Chapter 3, the stopping criterion of *Step 1(ii)* is changed to

$$\textbf{if} \ \ (\|\hat{d}_k\| \leq \epsilon) \ \ \textbf{stop},$$

where $\epsilon > 0$ is small. Finally, the details of the line search are also the same as described in Section 3.4.

In order to judge the efficiency of algorithm **FSQP′-MOC**, we ran the same numerical tests with two other algorithms differing only in the manner in which Ω_k and Ξ_k are updated. In the tables, the implementation of **FSQP′-MOC** just discussed is denoted NEW. A simple ϵ-active strategy was employed in the algorithm we call ϵ-ACT, i.e. we set $\Omega_k = \{ \, \omega \in \Omega \mid f(x_k, \omega) > F_\Omega(x) - \epsilon \, \}$ and $\Xi_k = \Xi_\epsilon(x_k)$ for all k, where $\epsilon = 0.1$. The algorithm of Chapter 3 was applied in algorithm FULL by simply setting $\Omega_k = \Omega$ and $\Xi_k = \Xi$, for all k. All three algorithms were set to stop when $\|d_k^0\| \leq 1 \times 10^{-4}$. A uniform discretization with 501 sample points was used in all cases. Problems `cw_2`, `cw_3`, and `cw_5` are borrowed from [10]. Problems with the prefix `oet` are from [45]. The problems from [45] are more naturally posed as mini-max problems. In order to also use them as constrained problems for Table 5.1 we used the trick discussed in Section 4.1 and added a variable.

The first two columns of the tables are self-explanatory. A description of the remaining columns is as follows. The third column, n, indicates the number of variables, while m_ℓ and m_n in the next two columns of Table 5.1 indicate the number of linear SIP constraints and nonlinear SIP constraints ($m_n = m - m_\ell$), respectively. Next, `NF` is the number of scalar objective function evaluations (i.e.

evaluation of $f(x)$ or some $f(x, \omega)$ for a given x and ω), `NG` is the number of "scalar" constraint function evaluations (i.e. evaluation of some $\phi_j(x, \xi)$ for a given x and ξ), and `IT` indicates the number of iterations required before the stopping criterion was satisfied. In Table 5.1, $f(x^*)$ indicates the value of the objective function at the final iterate, while in Table 5.2, $F(x^*)$ indicates the value of the maximum objective function at the final iterate. Finally, $\sum |\Xi_k|$ and $\sum |\Omega_k|$ are the sums over all iterations of the size of Ξ_k and Ω_k, respectively (they are equal to the number of gradient evaluations in the case of NEW and FULL), $|\Xi^*|$ and $|\Omega^*|$ are the sizes of Ξ_k and Ω_k at the final iterate, and `TIME` is the time of execution in seconds on a Sun Sparc 20 workstation. The * in the row for problem `oet_7` in Table 5.2 indicates that the algorithm failed to converge within 500 iterations.

A few conclusions may be drawn from the results. On average, NEW requires the most iterations to "converge" to a solution, whereas FULL requires the least. Of course, such behavior is expected since NEW uses a simpler QP model at each iteration. It is clear from comparing the results for $\sum |\Xi_k|$ or $\sum |\Omega_k|$ that NEW provides significant savings in the number of gradient evaluations and the size of the QP sub-problems. The savings for ϵ-ACT are not as dramatic. In almost all cases, comparing `TIME` of execution confirms that, indeed, NEW requires far less computational effort than either of the other two approaches. For problems in which gradient evaluations are expensive, the savings in time and computational effort would be even more dramatic than what is reported here.

PROB	ALGO	n	m_ℓ	m_n	NF	NG	IT	$f(x^*)$	$\|\Xi^*\|$	$\sum \|\Xi_k\|$	TIME
oet_1	NEW	2	0	1	18		18	5.382e-3	4	48	0.43
	ϵ-ACT				9		9	5.382e-3	224	1789	0.44
	FULL				6		6	5.382e-3	1002	6012	0.65
oet_2	NEW	2	0	1	4	4148	4	8.716e-2	3	17	0.14
	ϵ-ACT				8	9573	8	8.716e-2	557	1900	0.36
	FULL				4	4016	4	8.716e-2	1002	4008	0.44
oet_3	NEW	2	0	1	15		15	4.505e-3	4	86	0.38
	ϵ-ACT				8		8	4.505e-3	1002	3572	0.61
	FULL				6		6	4.505e-3	1002	6012	0.62
oet_4	NEW	2	0	1	18	22740	19	4.328e-3	5	92	0.49
	ϵ-ACT				16	20766	17	4.295e-3	1002	6180	1.12
	FULL				15	18585	16	4.296e-3	1002	16032	1.86
oet_5	NEW	2	0	1	46	54056	33	2.650e-3	4	175	0.99
	ϵ-ACT				28	34610	28	2.650e-3	1002	19825	3.21
	FULL				49	53890	36	2.650e-3	1002	36072	5.53
oet_6	NEW	2	0	1	19	25099	20	2.070e-3	5	119	0.62
	ϵ-ACT				22	24429	21	2.073e-3	1002	15466	3.04
	FULL				16	17595	15	2.073e-3	1002	15030	2.52
cw_2	NEW	2	0	1	5	2811	5	2.618	2	10	0.12
	ϵ-ACT				8	5530	7	2.618	501	1743	0.24
	FULL				5	3249	5	2.618	501	2505	0.30
cw_3	NEW	2	0	1	22	13868	25	5.335	2	48	0.35
	ϵ-ACT				17	12923	20	5.335	501	142	0.31
	FULL				22	13868	25	5.335	501	12525	0.96
cw_5	NEW	2	0	1	47		47	4.301	2	142	0.40
	ϵ-ACT				7		7	4.301	501	2001	0.25
	FULL				5		5	4.301	501	2505	0.28

Table 5.1: Numerical results for constrained problems with **FSQP′-MOC**.

PROB	ALGO	n	m_o	NF	IT	$F(x^*)$	$\lvert\Omega^*\rvert$	$\sum\lvert\Omega_k\rvert$	TIME
oet_1	NEW	2	2	11088	10	5.382e-1	4	45	0.29
	ϵ-ACT			6025	6	5.382e-1	224	1109	0.26
	FULL			6024	6	5.382e-1	1002	6012	0.56
oet_2	NEW	2	2	4017	4	8.717e-2	3	16	0.13
	ϵ-ACT			4017	4	8.717e-2	557	2080	0.27
	FULL			4017	4	8.717e-2	1002	4008	0.38
oet_3	NEW	3	2	7035	7	4.513e-3	6	33	0.20
	ϵ-ACT			9012	7	4.505e-3	1002	4222	0.48
	FULL			5023	5	4.505e-3	1002	5010	0.47
oet_4	NEW	3	2	11054	11	4.297e-3	6	51	0.28
	ϵ-ACT			13573	10	4.315e-3	1002	5357	0.66
	FULL			8038	8	4.302e-3	1002	8016	0.83
oet_5	NEW	4	2	29134	26	2.660e-3	4	150	0.70
	ϵ-ACT			43305	40	2.653e-3	1002	36469	4.97
	FULL			43207	43	2.652e-3	1002	40080	5.20
oet_6	NEW	4	2	17174	17	2.075e-3	7	107	0.52
	ϵ-ACT			66670	66	2.070e-3	1002	64655	10.38
	FULL			49493	49	2.070e-3	1002	49098	7.90
oet_7	NEW	6	2	192511	187	6.563e-5	7	1328	6.73
	ϵ-ACT					*			
	FULL			71004	70	7.351e-3	1002	70140	20.49

Table 5.2: Numerical results for mini-max problems with **FSQP′-MOC**.

Chapter 6

Implementation Details

6.1 Structure of the Implementation

The algorithms discussed in Chapters 3, 4, and 5 have been implemented in ANSI C [31] following the details discussed in Sections 3.4, 4.4, and 5.5. Note that the algorithm discussed in Chapter 5 applies equally well to problems with large sets of "sequentially related" objectives and/or constraints, not necessarily just discretized problems from SIP. As such, from this point forward, we will refer to such objectives and constraints as sequentially related (SR) instead of discretized SIP. The general problem tackled is

$$\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & \max\{\max_{i\in J} f_i(x), \max_{i\in J^{sr}} \max_{\omega\in\Omega^{f_i}} f_i(x,\omega)\} \\
\text{s.t.} \quad & x^\ell \le x \le x^u \\
& g_j(x) \le 0, \quad j \in I^n \\
& g_j(x,\xi) \le 0, \quad \xi \in \Xi^{g_j}, \;\; j \in I^{nsr} \\
& g_j(x) \equiv \langle a_{j-m_n}, x\rangle - b_{j-m_n} \le 0, \quad j \in I^a \\
& g_j(x,\xi) \equiv \langle a_{j-m_n}(\xi), x\rangle - b_{j-m_n}(\xi) \le 0, \quad \xi \in \Xi^{g_j}, \;\; j \in I^{asr},
\end{aligned}$$

where $J = \{1, \dots, p - p_{sr}\}$, $J^{sr} = \{p - p_{sr} + 1, \dots, p\}$, p is the total number of scalar objectives and sets of sequentially related objectives, p_{sr} is the number of sets of sequentially related objectives, $I^n = \{1, \dots, m_n - m_{nsr}\}$, $I^{nsr} = \{m_n - m_{nsr} + 1, \dots, m_n\}$, $I^a = \{m_n + 1, \dots, m - m_{\ell sr}\}$, $I^{asr} = \{m - m_{\ell sr} + 1, \dots, m\}$, m is the total number scalar constraints and sets of sequentially related constraints, m_n is the total number of nonlinear scalar constraints and sets of nonlinear sequentially related constraints, m_{nsr} is the number of sets of nonlinear sequentially related constraints, $m_{\ell sr}$ is the number of sets of affine sequentially related constraints, $a_{j-m_n} \in \mathbb{R}^n$, $b_{j-m_n} \in \mathbb{R}$, $j \in I^a$, $a_{j-m_n} : \Xi^{g_j} \to \mathbb{R}^n$, and $b_{j-m_n} : \Xi^{g_j} \to \mathbb{R}$, $j \in I^{asr}$.

The implementation, which we will call RFSQP (for *reduced* FSQP) follows the basic structure given in Figure 6.1. The user provides a main program (`main()` in the figure) which sets up the problem to be solved and calls `rfsqp()`. Whenever the algorithm requires objective and constraint values, and their gradients, it calls the user-defined functions which compute these quantities (`obj()`, `constr()`, `gradob`, and `gradcn()` in the figure). RFSQP calls the user-defined functions once for each scalar (objective or constraint) evaluation. This is in contrast to many optimization algorithm implementations which make one call to a user-defined function for all objective or constraint values. Note that the gradients need not be provided by the user, the implementation allows the user the option of letting it compute gradients via finite differences (see the end of this section). We allow the user to tune the algorithm parameters by changing them in the header file `param.h`. Finally, in order to solve the QP and LS subproblems, the implementation calls the solver QLD (`qld()` in the figure) due to Powell and Schittkowski [70]. The interface to the QP solver is designed so that a user could relatively easily use a QP solver other than QLD.

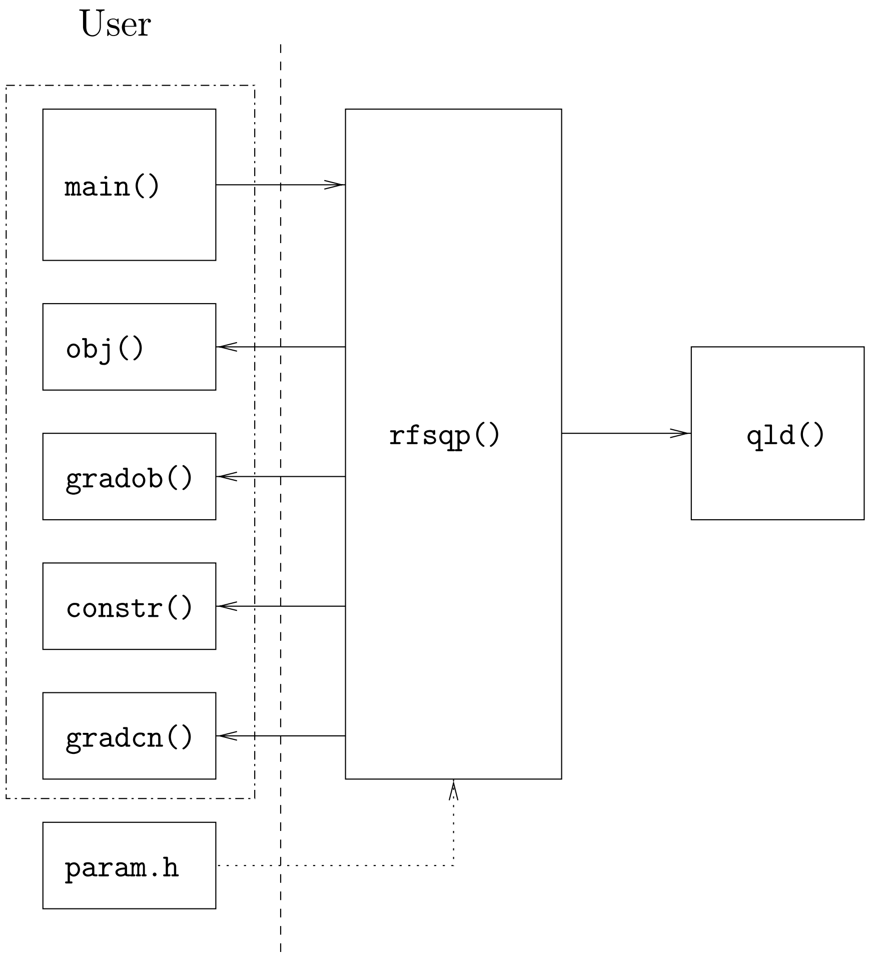

Figure 6.1: Structure of the implementation.

The calling sequence to `rfsqp()` in `main()` is

```
inform = rfsqp(nparam,nf,nineq,nineqn,nfsr,ncsrn,ncsrl,
          mesh_pts,iprint,miter,eps,bigbnd,x,bl,bu,f,g,
          lambda,obj,constr,gradob,gradcn);
```

The input and output parameters are defined as follows

`nparam` **(Input)** Number of free variables, i.e., n in the problem statement.

`nf` **(Input)** Number of objective functions, i.e. p in the problem statement. Note that one set of SR objectives counts as one objective.

`nineq` **(Input)** Total number of inequality constraints, i.e., m in the problem statement. Note that one set of SR constraints counts as one constraint.

`nineqn` **(Input)** Total number of affine inequality constraints, i.e. m_n in the problem statement.

`nfsr` **(Input)** Number of sets of SR objectives, i.e. p_{sr} in the problem statement. Must be less than or equal to `nf`.

`ncsrn` **(Input)** Number of sets of nonlinear SR constraints, i.e. m_{nsr} in the problem statement. Must be less than or equal to `nineqn`.

`ncsrl` **(Input)** Number of sets of affine SR constraints, i.e. $m_{\ell sr}$ in the problem statement. Must be less than or equal to `nineq - nineqn`.

`mesh_pts` **(Input)** Integer array containing the number of elements in each (*i*) SR objective set, (*ii*) nonlinear SR constraint set, and (*iii*) affine SR constraint set.

`iprint` **(Input)** Indicates amount of information to display during execution.

`iprint = 0` Display nothing.

`iprint = 1` Display all important output information after the final iteration.

`iprint = 2` Display same information as for `iprint = 1` at every iteration.

`iprint = 3` Dump most of the important internal variables at every iteration. Used for debugging.

`miter` **(Input)** Maximum number of iterations.

`eps` **(Input)** Stopping criterion, norm requirement on the search direction $\hat{d}_k$.

`bigbnd` **(Input)** Used in places where "infinity" is called for, e.g. for simple bounds in QP sub-problems where there are no simple bounds.

`x` **(Input/Output)** Double precision array which, on input, contains the user's initial guess, i.e. x_0, and, on output, contains the computed optimal solution.

`bl` **(Input)** Double precision array containing lower bounds on the variables x, i.e. x^ℓ from the problem statement.

`bu` **(Input)** Double precision array containing upper bounds on the variables x, i.e. x^u from the problem statement.

`f` **(Output)** Double precision array which, on output, contains the values of all (in the order specified by the problem statement) scalar objective functions at the solution.

`g` **(Output)** Double precision array which, on output, contains the values of all (in the order specified by the problem statement) scalar constraints at the solution.

`lambda` **(Output)** Double precision array which, on output, contains the values of all multiplier estimates in the order (i) simple bounds (`nparam` values since only `nparam` simple bounds could be active at the solution), (ii) objective functions, and (iii) constraints.

obj **(Input)** Pointer to a function computing the values of the objective functions.

```
void
obj(int nparam, int j, double *x, double *fj)
{
    /*
       for given j, assign to *fj the value of the
       (j+1)st objective evaluated at x
    */
   return;
}
```

Each member of a set of sequentially related objectives is assigned a unique value of j.

constr **(Input)** Pointer to a function computing the values of the constraint functions.

```
void
constr(int nparam, int j, double *x, double *gj)
{
    /*
       for given j, assign to *gj the value of the
       (j+1)st constraint evaluated at x
    */
   return;
}
```

Each member of a set of sequentially related constraints is assigned a unique value of j.

`gradob` **(Input)** Pointer to a function computing the gradient of the objective functions. Setting `gradob` = NULL causes RFSQP to use finite difference gradients.

```
void
gradob(int nparam, int j, double *x, double *gradfj)
{
   /*
      for i=1 to nparam assign to gradfj[i-1] the
      partial derivative of the (j+1)st objective
      with respect to the ith parameter evaluated at x
   */
   return;
}
```

Objective indexing must follow exactly that in `obj()`.

`gradcn` **(Input)** Pointer to a function computing the gradient of the constraint functions. Setting `gradcn` = NULL causes RFSQP to use finite difference gradients.

```
void
gradcn(int nparam, int j, double *x, double *gradgj)
{
   /*
      for i=1 to nparam assign to gradgj[i-1] the
      partial derivative of the (j+1)st constraint
      with respect to the ith parameter evaluated at x
   */
   return;
}
```

Constraint indexing must follow exactly that in `constr()`.

`inform` **(Output)** Indicates status of execution.

`inform = 0` Normal termination.

`inform = 1` Failure in the QP solver.

`inform = 2` Failure in the line search. The step size t_k is smaller than the machine precision.

`inform = 3` Maximum number of iterations `maxit` reached.

`inform = 4` Unable to generate a feasible initial point for nonlinear constraints (see Section 6.2).

`inform = 5` Unable to generate a feasible initial point for affine constraints (see Section 6.2).

In the event that the user passes a NULL pointer for `gradob()` and/or `gradcn()`, the implementation will compute the gradients via forward finite differences. At iteration k let x_k^i denote the ith component of the iterate x_k. Define the perturbations $\delta_i \in \mathbb{R}^n$, $i = 1, \ldots, n$, as

$$\delta_i^j \triangleq \begin{cases} \sqrt{\epsilon_m} \cdot \max\{1,\ |x_k^i|\}, & j = i, \\ 0, & \text{otherwise}, \end{cases}$$

where, as usual, ϵ_m denotes the machine precision. Then, for an objective f_j, we use the approximation

$$\frac{\partial f_j}{\partial x^i} \approx \frac{f_j(x_k + \delta_i) - f_j(x_k)}{\|\delta_i\|}.$$

A similar expression is used for constraints.

6.2 Infeasible Initial Point

Note that in all of the algorithm descriptions thus far we have assumed that the user specifies a feasible initial guess, i.e. $x_0 \in X$. This is a restrictive assumption, in general, and an implementation should be able to deal with an infeasible initial guess. In this section we discuss the approach used in RFSQP to generate a feasible initial point. We follow the approach used in CFSQP/FFSQP [36, 79].

Let x_0 denote the initial guess provided by the user. The first step is to check all affine constraints to see if they are satisfied. If not, then we solve the following convex QP

$$\begin{array}{ll}\min\limits_{v\in\mathbb{R}^n} & \langle v,v\rangle \\ \text{s.t.} & x^\ell \le x_0+v \le x^u \\ & \langle a_{j-m_n}, x_0+v\rangle - b_{j-m_n} \le 0, \quad j\in I^a \\ & \langle a_{j-m_n}(\xi), x_0+v\rangle - b_{j-m_n}(\xi) \le 0, \quad \xi\in\Xi^{g_j}, \quad j\in I^{asr}.\end{array}$$

Note that this QP is consistent if, and only if, a point exists which satisfies all of the affine constraints for the original problem. If so, then the unique solution of the QP is the smallest perturbation of the initial guess provided by the user which is feasible for the affine constraints. Letting v^* denote the solution, we set $x_0' = x_0 + v^*$.

The next step is to check whether x_0' satisfies all nonlinear constraints. If so, then we may proceed with x_0' as the initial point. Otherwise, we iterate (using the algorithms presented in this dissertation) on the problem

$$\begin{array}{ll}\min\limits_{x\in\mathbb{R}^n} & \max\{\max\limits_{i\in I^n} g_i(x), \max\limits_{i\in I^{nsr}} \max\limits_{\xi\in\Xi^{g_i}} g_i(x,\xi)\} \\ \text{s.t.} & x^\ell \le x \le x^u \\ & \langle a_{j-m_n}, x\rangle - b_{j-m_n} \le 0, \quad j\in I^a \\ & \langle a_{j-m_n}(\xi), x\rangle - b_{j-m_n}(\xi) \le 0, \quad \xi\in\Xi^{g_j}, \quad j\in I^{asr},\end{array}$$

using x_0' as the initial point. Of course, it is not necessary to iterate on this problem until a KKT point is detected. Instead, after the line search in each iteration k' is performed, we check to see whether

$$G(x_{k'+1}) \stackrel{\Delta}{=} \max\{\max_{i\in I^n} g_i(x_{k'+1}), \max_{i\in I^{nsr}} \max_{\xi\in\Xi^{g_i}} g_i(x_{k'+1},\xi)\} \leq 0.$$

If so, then we immediately stop, set $x_0'' = x_{k'+1}$, and begin iterating on the original problem using x_0'' as the *feasible* initial point.

6.3 BFGS Updates for Cholesky Factors

At each iteration of our algorithms, in order to solve the QP and the two least squares problems, the solver(s) must typically perform a Cholesky decomposition (see [17]) of the Hessian estimate H_k, i.e. compute

$$H_k = R_k^T R_k,$$

where $R_k \in \mathbb{R}^{n\times n}$ is upper triangular. Of course, repeating this procedure (which requires $O(n^3)$ operations) three times is wasteful, especially for problems where n is large. Thus, it would be ideal if we could maintain and update the Cholesky factor R_k instead of H_k itself. Several authors (see, e.g., [16, 14]) have proposed schemes for performing rank-two updates (such as the BFGS update given in Section 3.4) on the Cholesky factors of a positive definite matrix. In the implementation RFSQP we use the approach from [14], which we briefly review here.

We will actually update the equivalent $L_k D_k L_k^T$ factorization of H_k, where L_k is lower triangular with all ones on the main diagonal and D_k is diagonal. From this factorization, it is a trivial matter to obtain the Cholesky factor, specifically

$$R_k = D_k^{1/2} L_k^T.$$

Recall that we are interested in the rank-two update

$$L_{k+1}D_{k+1}L_{k+1}^T = L_kD_kL_k^T - uu^T + vv^T,$$

where, from Section 3.4,

$$u \triangleq \frac{H_k\delta_{k+1}}{\sqrt{\delta_{k+1}^T H_k \delta_{k+1}}}, \qquad v \triangleq \frac{\xi_{k+1}}{\sqrt{\delta_{k+1}^T \xi_{k+1}}}.$$

These vectors may be efficiently computed from the factorizations of H_k directly. For ease of notation we will dispense with the subscript k and let R, L, and D denote the respective matrices at the current iteration and R_+, L_+, and D_+ denote the updated matrices. The update is completed in two major steps, one for each dyad, positive and negative. First, we obtain $\bar{L}$ and $\bar{D}$ from L and D using the positive correction vv^T. Let $L = [\ell_{i,j}]$ and $D = \text{diag}\{d_1, \ldots, d_n\}$, similarly for $\bar{L}$ and $\bar{D}$. The following procedure is from [14].

```
set $\tau_0 = 1$, $\nu^1 = v$
for $j = 1, \ldots, n$ do {
    $p_j = \nu_j^j$
    $\tau_j = \tau_{j-1} + p_j^2/d_j$
    $\bar{d}_j = d_j\tau_j/\tau_{j-1}$
    $\beta_j = p_j/(d_j\tau_j)$
    for $r = j+1, \ldots, n$ do {
        $\nu_r^{j+1} = \nu_r^j - p_j\ell_{r,j}$
        $\bar{\ell}_{r,j} = \ell_{r,j} + \beta_j\nu_r^{j+1}$
    }
}
```

The update for the negative correction $-uu^T$ is a more involved since care must be taken to ensure that rounding errors don't cause any of the elements of D_+ to become zero or negative (in which case H_{k+1} would not be positive definite). Let $L_+ = [\ell^+_{i,j}]$ and $D_+ = \text{diag}\{d^+_1, \ldots, d^+_n\}$. The following procedure is also from [14].

```
**solve** $\bar{L}p = u$ and **set** $\tau_{n+1} = 1 - p^T \bar{D}^{-1} p$
**if** $(\tau_{n+1} \leq \epsilon_m)$ **then set** $\tau_{n+1} = \epsilon_m$
**for** $j = n, \ldots, 1$ **do** {
    $\tau_j = \tau_{j+1} + p_j^2 / \bar{d}_j$
    $d_j^+ = \bar{d}_j \tau_{j+1} / \tau_j$
    $\beta_j = -p_j / (\bar{d}_j \tau_{j+1})$
    $u_j^j = p_j$
    **for** $r = j+1, \ldots, n$ **do** {
        $\ell^+_{r,j} = \bar{\ell}_{r,j} + \beta_j u_r^{j+1}$
        $u_r^j = u_r^{j+1} + p_j \bar{\ell}_{r,j}$
    }
}
```

Note that the first step requires the solution of the linear system $\bar{L}p = u$. As $\bar{L}$ is lower triangular, solving this system is just a simple matter of forward substitution. Finally, R_+ is readily computed from L_+ and D_+. Thus, we have a procedure for updating the Cholesky factors of H_k. Of course, this update is more expensive computationally than directly updating H_k, but the savings gained by not having to perform Cholesky factorizations in each of the three sub-problems outweighs the increase in computation required for the update.

Our implementation RFSQP gives the user the option (through the header file `param.h`) of either updating H_k directly, or the Cholesky factor R_k.

6.4 A Note on the Linear Algebra

It is not difficult to see that solving the least squares problems for $\widehat{d^0}$ and $\tilde{d}$ is equivalent to solving linear systems of the form

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{pmatrix} d \\ \lambda \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{6.1}$$

In this section we discuss an efficient method due to Gay, Overton, and Wright [13] for solving such systems. Note that this method has not yet been implemented in RFSQP.

The first step is to perform a QR decomposition (see, e.g., [17]) of A^T. Suppose that $0 < H = H^T \in \mathbb{R}^{n\times n}$ and $A \in \mathbb{R}^{m\times n}$. Then

$$A^T = QR = [Q_1 \;\; Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where $Q \in \mathbb{R}^{n\times n}$ is orthogonal, $R \in \mathbb{R}^{n\times m}$, $Q_1 \in \mathbb{R}^{n\times m}$, and $R_1 \in \mathbb{R}^{m\times m}$ is upper triangular. We may write $d = Q_1 d_1 + Q_2 d_2$. Hence

$$\begin{aligned} Ad &= AQ_1 d_1 + AQ_2 d_2 \\ &= [R_1^T \;\; 0] \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (Q_1 d_1 + Q_2 d_2) \\ &= R_1^T d_1. \end{aligned}$$

Thus, d_1 may be obtained by forward substitution from the triangular system

$$R_1^T d_1 = r_2. \tag{6.2}$$

Now, from the first equation in (6.1), we have

$$\begin{aligned} r_1 &= Hd + A^T\lambda \\ &= HQ_1d_1 + HQ_2d_2 + A^T\lambda. \end{aligned}$$

Multiplying by Q_2^T and noting that $Q_2^TA^T = 0$,

$$Q_2^Tr_1 = Q_2^THQ_1d_1 + Q_2^THQ_2d_2.$$

Defining $v = Q_2^Tr_1 - Q_2^THQ_1d_1$ and substituting the Cholesky factorization $H = R^TR$ (which we already have, see Section 6.3), gives

$$Q_2^TR^TRQ_2d_2 = v.$$

We would like to solve this equation for d_2. To begin, define $z = R^TRQ_2d_2$, and consider solving $Q_2^Tz = v$. Writing $Q = [q_1, \dots, q_n]$, where $q_i \in \mathbb{R}^n$, $i = 1, \dots n$, it is not difficult to show that

$$z = \sum_{i=1}^{n-m} v_iq_{m+i}. \tag{6.3}$$

We may then use forward substitution to solve the triangular system

$$R^Ty = z \tag{6.4}$$

for y, and the immediately use back substitution to solve

$$Rw = y \tag{6.5}$$

for w. Finally, we are left with $w = Q_2d_2$, hence $d_2 = Q_2^Tw$ and we have $d = Q_1d_1 + Q_2d_2 = Q_1d_1 + w$. Now, in order to compute λ, note that $A^T\lambda = r_1 - Hd$. Thus, substituting the QR decomposition, we have the triangular system

$$R_1\lambda = Q_1^T(r_1 - Hd), \tag{6.6}$$

which may be solved via back substitution to obtain λ. Summing up, solving (6.1) involves the following steps

1. Obtain QR decomposition of A^T.

2. Use forward substitution to solve (6.2) for d_1.

3. Form z according to (6.3).

4. Use forward substitution to solve (6.4) for y.

5. Use back substitution to solve (6.5) for w.

6. Form $d = Q_1 d_1 + w$.

7. Use back substitution to solve (6.6) for λ.

For large problems the QR decomposition dominates the other steps in terms of computational cost. The two least squares problems for $\widehat{d^0}$ and $\tilde{d}$ will likely have similar, or identical, "A" matrices (H is the Hessian estimate H_k in both cases). When they are different, it is typically only by the addition and/or deletion of a few rows. In this case, instead of computing the QR factorization from scratch each time, it may make more sense to employ updating and downdating procedures as described in Section 12.6 of [17].

6.5 Full QP for the Maratos Correction

There may be times where it would be preferable to use a full QP model for $\tilde{d}$, as is done in [51], instead of the least squares problem we use here. This may be the case, for example, if function evaluations are very expensive. In such a situation, it is important to use the best possible model of the problem at each iteration in order to (i) reduce the total number of iterations, and (ii) increase the likelihood that a full step of one will be accepted in the line search. Both properties have the effect of reducing the total number of function evaluations.

The implementation RFSQP allows the user the option of using a full QP for $\tilde{d}$. Still, not all objectives and constraints need to be included in the QP, only those which are active for $\widehat{QP}$ and those which are close to active for the original problem. Define

$$F(x) \overset{\Delta}{=} \max\{\max_{i\in J} f_i(x), \max_{i\in J^{sr}} \max_{\omega\in\Omega^{f_i}} f_i(x,\omega)\}.$$

Now define the index sets of objectives which were active for $\widehat{QP}$,

$$\begin{aligned} \hat{J}_k &= \{\ j \in J \mid f_j(x_k) + \langle \nabla f_j(x_k), \hat{d}_k\rangle - F(x_k) - \hat{\gamma}_k > -\sqrt{\epsilon_m}\ \}, \\ \hat{\Omega}_k^{f_j} &= \{\ \omega \in \Omega_k^{f_j} \mid f_j(x_k,\omega) + \langle \nabla f_j(x_k,\omega), \hat{d}_k\rangle - F(x_k) - \hat{\gamma}_k > -\sqrt{\epsilon_m}\ \}, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j \in J^{sr}, \end{aligned}$$

and constraints which were active for $\widehat{QP}$,

$$\begin{aligned} \hat{I}_k^n &= \{\ j \in I^n \mid g_j(x_k) + \langle \nabla g_j(x_k), \hat{d}_k\rangle - \hat{\gamma}_k \cdot \eta_k^j > -\sqrt{\epsilon_m}\ \}, \\ \hat{\Xi}_k^{n,g_j} &= \{\ \xi \in \Xi_k^{g_j} \mid g_j(x_k,\xi) + \langle \nabla g_j(x_k,\xi), \hat{d}_k\rangle - \hat{\gamma}_k \cdot \eta_k^j > -\sqrt{\epsilon_m}\ \}, \quad j \in I^{nsr}, \\ \hat{I}_k^a &= \{\ j \in I^a \mid \langle a_{j-m_n}, x_k + \hat{d}_k\rangle + b_{j-m_n} > -\sqrt{\epsilon_m}\ \}, \\ \hat{\Xi}_k^{a,g_j} &= \{\ \xi \in \Xi_k^{g_j} \mid \langle a_{j-m_n}(\xi), x_k + \hat{d}_k\rangle + b_{j-m_n}(\xi) > -\sqrt{\epsilon_m}\ \}, \quad j \in I^{asr}. \end{aligned}$$

Following [36, 79], let f_k^b be the value at x_k of the first objective which has a positive multiplier in $\widehat{QP}$ and let ∇f_k^b denote its gradient at x_k. Now define the index sets of objectives which are "close" to active for the original problem,

$$\begin{aligned} \bar{J}_k &= \{\ j \in J \mid |f_j(x_k) - f_k^b| \leq 0.2\|\hat{d}_k\| \cdot \|\nabla f_j(x_k) - \nabla f_k^b\|\}, \\ \bar{\Omega}_k^{f_j} &= \{\ \omega \in \Omega_k^{f_j} \mid |f_j(x_k,\omega) - f_k^b| \leq 0.2\|\hat{d}_k\| \cdot \|\nabla f_j(x_k,\omega) - \nabla f_k^b\|\}, \quad j \in J^{sr}. \end{aligned}$$

Finally, define the set of indices of constraints which are "close" to active for the original problem,

$$
\begin{aligned}
\bar{I}^n_k &= \{\ j \in I^n \mid |g_j(x_k)| \leq 0.2\|\hat{d}_k\|\cdot\|\nabla g_j(x_k)\|\ \},\\
\bar{\Xi}^{n,g_j}_k &= \{\ \xi \in \Xi^{g_j}_k \mid |g_j(x_k,\xi)| \leq 0.2\|\hat{d}_k\|\cdot\|\nabla g_j(x_k,\xi)\|\ \}, \quad j \in I^{nsr},\\
\bar{I}^a_k &= \{\ j \in I^a \mid |\langle a_{j-m_n}, x_k\rangle + b_{j-m_n}| \leq 0.2\|\hat{d}_k\|\cdot\|a_{j-m_n}\|\ \},\\
\bar{\Xi}^{a,g_j}_k &= \{\ \xi \in \Xi^{g_j}_k \mid |\langle a_{j-m_n}(\xi), x_k\rangle + b_{j-m_n}(\xi)| \leq 0.2\|\hat{d}_k\|\cdot\|a_{j-m_n}(\xi)\|\ \},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j \in I^{asr}.
\end{aligned}
$$

Let $\tilde{F}_k$ be the maximum value of all objectives which we will be used in the computation of $\tilde{d}$, evaluated at $x_k + \hat{d}_k$, i.e.

$$
\tilde{F}_k \triangleq \max\{\max_{i\in\hat{J}_k\cup\bar{J}_k} f_i(x_k+\hat{d}_k),\quad \max_{i\in J^{sr}}\ \max_{\omega\in\hat{\Omega}^{f_i}_k\cup\bar{\Omega}^{f_i}_k} f_i(x_k+\hat{d}_k,\omega)\}.
$$

The full QP used to compute $\tilde{d}_k$ in the RFSQP implementation (when the user chooses not to use the least squares option) is as follows,

$$
\begin{aligned}
\min_{\tilde{d}\in\mathbb{R}^n,\tilde{\gamma}\in\mathbb{R}} \quad & \langle \hat{d}_k + \tilde{d}, H_k(\hat{d}_k + \tilde{d})\rangle + \tilde{\gamma}\\
\text{s.t.} \quad & x^\ell \leq x_k + \hat{d}_k + \tilde{d} \leq x^u\\
& f_j(x_k+\hat{d}_k) + \langle \nabla f_j(x_k), \tilde{d}\rangle \leq \tilde{F}_k + \tilde{\gamma}, \quad j \in \hat{J}_k \cup \bar{J}_k,\\
& f_j(x_k+\hat{d}_k,\omega) + \langle \nabla f_j(x_k,\omega), \tilde{d}\rangle \leq \tilde{F}_k + \tilde{\gamma}, \quad \omega \in \hat{\Omega}^{f_j}_k \cup \bar{\Omega}^{f_j}_k,\ j \in J^{sr},\\
& g_j(x_k+\hat{d}_k) + \langle \nabla g_j(x_k), \tilde{d}\rangle \leq -\min\{10^{-2}\|\hat{d}_k\|, \|\hat{d}_k\|^\tau\}, \quad j \in \hat{I}^n_k \cup \bar{I}^n_k,\\
& g_j(x_k+\hat{d}_k,\xi) + \langle \nabla g_j(x_k,\xi), \tilde{d}\rangle \leq -\min\{10^{-2}\|\hat{d}_k\|, \|\hat{d}_k\|^\tau\},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \xi \in \hat{\Xi}^{g_j}_k \cup \bar{\Xi}^{g_j}_k, \quad j \in I^{nsr}\\
& \langle a_{j-m_n}, x_k + \hat{d}_k + \tilde{d}\rangle - b_{j-m_n} \leq 0, \quad j \in \hat{I}^a_k \cup \bar{I}^a_k\\
& \langle a_{j-m_n}(\xi), x_k + \hat{d}_k + \tilde{d}\rangle - b_{j-m_n}(\xi) \leq 0, \quad \xi \in \hat{\Xi}^{g_j}_k \cup \bar{\Xi}^{g_j}_k, \quad j \in I^{asr},
\end{aligned}
$$

where $\tau \in (2,3)$.

Chapter 7

Application to Engineering Design

7.1 Introduction

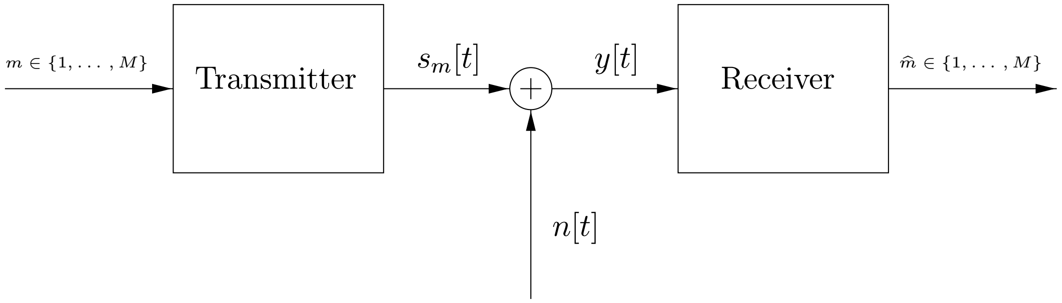

Figure 7.1: Model of a communication system.

Consider the simple communication system model shown in Figure 7.1. The goal is to transmit one of M possible symbols, i.e. an M-ary signaling system, over a memoryless additive noise channel. We will assume all signals are discrete-time with T samples. The transmitter assigns a unique signal $s_m : \{1, \dots, T\} \to \mathbb{R}$ to each symbol $m \in \{1, \dots, M\}$. It is this signal that is sent through the channel. At the other end, the received signal is

$$y[t] = s_m[t] + n[t], \quad t = 1, \dots, T,$$

where $n : \{1, \ldots, T\} \to \mathbb{R}$ is a noise process, and the job of the receiver is to decide which symbol was transmitted. Our goal is to apply the algorithms developed in this dissertation to design a set of signals s_m, $m = 1, \ldots, M$, which maximizes, subject to constraints on the signals, the probability of a correct decision by the receiver given a particular channel noise distribution.

Of course, in order to design an optimal signal set, the action of the channel and the receiver must be completely specified. For the channel, we assume the noise process is independent and identically distributed (iid) with distribution p_N. Further, we assume that the noise process is independent of the symbol being transmitted. Our detection problem falls into the class of M-ary Bayesian hypothesis testing problems where, for $m = 1, \ldots, M$, the hypotheses are defined as follows,

$$H_m : \quad y[t] = s_m[t] + n[t], \qquad t = 1, \ldots, T.$$

To simplify notation, define the received signal vector

$$\mathbf{y} \triangleq (y[1], \ldots, y[T])^T.$$

Finally, it is assumed that the receiver was designed using the minimum average probability of error criterion (or the uniform cost criterion). It is well known that (see, e.g., Section IV.B of [59]), under our assumptions, the optimal receiver is the *maximum a posteriori probability* (MAP) detector. Specifically, the optimal receiver chooses

$$\hat{m}(\mathbf{y}) = \arg\max \{ \, p(H_m | \, \mathbf{y}) \mid m = 1, \ldots, M \, \},$$

i.e. the hypothesis with the largest probability given the observation $\mathbf{y}$.

Clearly, the receiver will make an error if hypothesis H_m is true, but

$$p(H_{m'} | \, \mathbf{y}) > p(H_m | \, \mathbf{y}),$$

for some $m' \neq m$. Thus, the probability of a correct decision under hypothesis H_m is

$$p(\{\text{correct decision}\} \mid H_m) = p\,(\ \{\ p(H_m|\ \mathbf{y}) > p(H_{m'}|\ \mathbf{y}),\ \ \forall m' \neq m\} \mid H_m)$$
$$= p\left(\ \left\{\ln \frac{p(H_m|\ \mathbf{y})}{p(H_{m'}|\ \mathbf{y})} > 0, \quad \forall m' \neq m\right\}\ \middle|\ H_m\ \right),$$

where, in order to put things in terms of the familiar log-likelihood ratio, we have assumed $p(H_{m'}|\ \mathbf{y}) > 0$ for all $\mathbf{y}$, $m' \in \{1, \ldots, M\}$. For the signal set design problem considered here, no knowledge of the prior distribution on the hypotheses H_m, $m = 1, \ldots, M$, will be assumed . Of course, the conditional distribution $p(H_m \mid \mathbf{y})$ is known since, given a signal set, this distribution is completely determined by the distribution on the channel noise. Specifically, in view of our assumptions,

$$p(H_m \mid \mathbf{y}) = \sum_{t=1}^{T} p_N(y[t] - s_m[t]).$$

If the prior distribution were known, the quantity to be maximized could be expanded as

$$p(\{\text{correct decision}\}) = \sum_{m=1}^{M} p(\{\text{correct decision}\} \mid H_m) \cdot p(H_m).$$

As $p(H_m)$ is not assumed to be known, the worst-case prior distribution will be used to compute $p(\{\text{correct decision}\})$ for any given signal set. In particular, let

$$\mathcal{S} \triangleq \left\{\ \gamma \in \mathbb{R}^M\ \middle|\ \sum_{m=1}^{M} \gamma_m = 1, \ \ \gamma_m \geq 0, \ \ m = 1, \ldots, M\ \right\}.$$

The goal will be to find signal sets which maximize

$$\min_{\gamma \in \mathcal{S}} \sum_{m=1}^{M} p(\{\text{correct decision}\} \mid H_m) \cdot \gamma_m.$$

It is not difficult to show that this is equivalent to maximizing

$$\min_{m\in\{1,\ldots,M\}} p(\ \{\text{correct decision}\} \mid H_m). \tag{7.1}$$

A standard assumption in transmitter design is that the signals are restricted to be of the form

$$s_m[t] \triangleq \sum_{k=1}^{K} \alpha^{m,k}\phi_k[t], \tag{7.2}$$

where $\phi_k : \{1,\ldots,T\} \to \mathbb{R}$, $k = 1,\ldots,K$, are given basis functions and $\alpha^{m,k} \in \mathbb{R}$, $m = 1,\ldots,M$, $k = 1,\ldots,K$, are the free parameters. Finally, due to power limitations in the transmitter, the signals are forced to satisfy some type of power constraint, either peak amplitude or average energy. In this chapter, we will assume a peak amplitude constraint, i.e.

$$|s_m[t]| \le C, \quad m = 1,\ldots,M, \quad t = 1,\ldots,T, \tag{7.3}$$

where $C > 0$ is given. Note that we could just as easily have considered an average energy constraint in our formulation. Our design problem is thus reduced to choosing parameters $\alpha^{m,k}$ in order to maximize (7.1), subject to the constraints (7.3).

7.2 The Optimization Problem

In this section we go through the steps of framing the problem discussed in the previous section in such a way that it may be efficiently solved using the algorithms developed in this dissertation. The design of optimal signal sets under the assumption of Gaussian noise has been well studied (see, e.g., [63]). In fact, a gradient-based first-order algorithm was developed and analyzed in [12] for the case of Gaussian noise, $K = 2$ basis functions, and an average energy constraint on the signals. The performance of optimal detectors in the presence of non-Gaussian noise (as a function of signal set choice) was first studied by Johnson and Orsak in [29]. It was shown in [29] that the dependence of detector performance on the signal set is related to the *Kullback-Leibler* (KL) distance between distributions for the various hypotheses. Based on this work, Gockenbach and Kearsley [15] proposed the nonlinear programming (NLP) formulation of the signal set design problem which is considered here.

Given our assumptions on the noise process, the log-likelihood ratio may be written

$$\ln \frac{p(H_m|\, \mathbf{y})}{p(H_{m'}|\, \mathbf{y})} = \sum_{t=1}^{T} \ln \frac{p(H_m|\, y[t])}{p(H_{m'}|\, y[t])}.$$

Note that, since randomness only enters the received signal through the additive noise process, when hypothesis H_m is true, the receiver computes

$$p(H_m|\, y[t]) = p_N(n[t]),$$

and, for $m' \neq m$,

$$p(H_{m'}|\, y[t]) = p_N(n[t] + (s_{m'}[t] - s_m[t])).$$

Thus, upon substitution, the statistic of interest to us is

$$\ln \frac{p(H_m|\, \mathbf{y})}{p(H_{m'}|\, \mathbf{y})} = \sum_{t=1}^{T} \ln \left(\frac{p_N(n[t])}{p_N(n[t] + (s_{m'}[t] - s_m[t]))} \right). \tag{7.4}$$

Now, assuming the variance of the statistic (7.4) does not change as we vary $m' \neq m$, maximizing $p(\{\text{correct decision}\} \mid H_m)$ is equivalent to maximizing the expected value of the statistic (7.4) for each $m' \neq m$. That is, under hypothesis H_m, the probability of correctly choosing H_m is maximized if we maximize

$$\min_{m' \neq m} E \left\{ \sum_{t=1}^{T} \ln \left(\frac{p_N(n[t])}{p_N(n[t] + (s_{m'}[t] - s_m[t]))} \right) \;\middle|\; H_m \right\}.$$

Thus, the objective function for the signal design problem considered here is

$$\min_{m\in\{1,\dots,M\}} \min_{m'\neq m} E\left\{ \sum_{t=1}^{T} \ln\left(\frac{p_N(n[t])}{p_N(n[t]+(s_{m'}[t]-s_m[t]))} \right) \;\middle|\; H_m \right\}.$$

Define the function $K_N : \mathbb{R} \to \mathbb{R}$ as

$$K_N(\delta) \triangleq \int_{\mathbb{R}} \ln\left(\frac{p_N(\tau)}{p_N(\tau+\delta)} \right) p_N(\tau)d\tau,$$

i.e. the KL distance between the noise distribution and the noise distribution shifted by $-\delta$. Note that if we assume a symmetric distribution for the noise (this is not a restrictive assumption), then $K_N(\cdot)$ will be an even function. It is not difficult to see that

$$E\left\{ \sum_{t=1}^{T} \ln\left(\frac{p_N(n[t])}{p_N(n[t]+(s_{m'}[t]-s_m[t]))} \right) \;\middle|\; H_m \right\} = \sum_{t=1}^{T} K_N(s_{m'}[t]-s_m[t]).$$

Define

$$\alpha \triangleq (\alpha^{1,1},\dots,\alpha^{1,K},\dots,\alpha^{M,1},\dots,\alpha^{M,K}) \in \mathbb{R}^{MK}.$$

Substituting the expansion (7.2), we see that, under our assumptions, the signal set design problem is equivalent to solving the optimization problem

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^{MK}} \quad & \max\left\{ -\sum_{t=1}^{T} K_N\left(\sum_{k=1}^{K} (\alpha^{m',k}-\alpha^{m,k})\phi_k[t] \right) \;\middle|\; {\scriptstyle m,m'\in\{1,\dots,M\},\ m'>m} \right\} \\ \text{s.t.} \quad & \left(\sum_{k=1}^{K} \alpha^{m,k}\phi_k[t] \right)^2 \le C^2, \quad m=1,\dots,M, \quad t=1,\dots,T. \end{aligned} \tag{SS}$$

It is only necessary to consider $m' > m$ since $K_N(\cdot)$ is an even function.

Note that (SS) may be solved by the constrained mini-max algorithm of Chapter 4. It turns out, though, that it is better to use the constrained mini-max algorithm of Chapter 5 (Section 5.4, Algorithm **FSQP′-MOC**) due to the possibly large number of objectives and constraints. Using the notation of

Section 6.1 for the general problem tackled by the implementation RFSQP, let $J = \emptyset$, $J^{sr} = \{1\}$, $\Omega^{f_1} = \{1, \ldots, M(M-1)/2\}$, and $\Xi^{g_1} = \{1, \ldots, MT\}$. Define the mappings

$$\begin{aligned} m_1 &: \Omega^{f_1} \to \{1, \ldots, M\}, \\ m_2 &: \Omega^{f_1} \to \{1, \ldots, M\}, \\ m &: \Xi^{g_1} \to \{1, \ldots, M\}, \\ t &: \Xi^{g_1} \to \{1, \ldots, T\}, \end{aligned}$$

in any way so that $(m_1(\omega), m_2(\omega))$ is a one-to-one mapping from Ω^{f_1} onto $\{\ (m', m'') \mid m', m'' \in \{1, \ldots, M\},\ m'' > m'\ \}$, and $(m(\xi), t(\xi))$ is a one-to-one mapping from Ξ^{g_1} onto $\{\ (m', t') \mid m' \in \{1, \ldots, M\},\ t' \in \{1, \ldots, T\}\ \}$. Such mappings are not difficult to construct. Now define

$$f_1(\alpha, \omega) \triangleq -\sum_{t=1}^{T} K_N \left(\sum_{k=1}^{K} (\alpha^{m_2(\omega),k} - \alpha^{m_1(\omega),k}) \phi_k[t] \right), \quad \omega \in \Omega^{f_1},$$

and

$$g_1(\alpha, \xi) \triangleq \left(\sum_{k=1}^{K} \alpha^{m(\xi),k} \phi_k[t(\xi)] \right)^2 - C^2, \quad \xi \in \Xi^{g_1}.$$

Simple bounds on the variables α are defined in Section 7.3. Thus, letting $n = MK$, $m = m_n = m_{nsr} = 1$, and $m_{\ell sr} = 0$, the problem is completely specified in a form which can be tackled by the implementation RFSQP. C code which computes the objective and constraint function values is included in Appendix A.

7.3 Global Algorithms

Problem (SS) is an ideal application for the algorithms developed in this dissertation. To begin with, there are few algorithms available to directly handle the constrained mini-max problem. At first glance it may seem as though there is no reason to require feasible iterates for (SS). In fact, feasible iterates are desirable, but for an "algorithmic" reason instead of an application-oriented one. Specifically, it was observed in [15] that outside of the feasible region, the linearized constraints for problem (SS) are often inconsistent, i.e. no feasible solution exists. Of course, with feasible iterates, the linearized constraints are always consistent and the solutions of the QP sub-problems are always well-defined. For practical instances of the problem, the number of objective functions and nonlinear constraints is large, which makes the problem an excellent candidate for the application of Algorithm **FSQP′-MOC**. Finally, we note that Algorithm **FSQP′-MOC** is preferable to CFSQP [36] for solving (SS) because function evaluations are relatively cheap and are dominated by the computational cost of generating a new iterate.

The only difficulty in applying Algorithm **FSQP′-MOC** is that problem (SS) has many local solutions which may prevent convergence to a *global* solution. In an attempt to overcome this problem, we will use a stochastic two-phase method (see, e.g., [4]) where random initial points are generated in the feasible region and Algorithm **FSQP′-MOC**, the local method, is repeatedly applied to a subset of these points. Such an approach may be thought of as simply a "smart" way of generating many initial points for our fast local algorithm with the hopes of eventually identifying a global solution. Specifically, we will use the Multi-Level Single Linkage (**MLSL**) approach [4, 30], which is known to find, with probability one, all local minima (hence the global minima) in a finite number of iterations.

Let $\mathcal{M}$ denote the cumulative set of local minimizers identified by the **MLSL** algorithm. At iteration ℓ, for some integer $N > 0$ fixed, we generate N points

$\alpha_{(\ell-1)N+1}, \ldots, \alpha_{\ell N}$ distributed uniformly over the feasible set X for (SS). For each of the points $\alpha_i \in \{\alpha_1, \ldots, \alpha_{\ell N}\}$ we check to see if there is another point α_j within a "critical distance" r_ℓ of α_i which also has a smaller objective value. If not, then the local algorithm **FSQP′-MOC** is applied with initial point α_i and the computed local minimizer is added to the set $\mathcal{M}$. After all points are checked, r_ℓ is updated, ℓ is set to $\ell+1$ and the process is repeated. At any given iteration, the local maximizer with the smallest objective value is our current estimate of the global solution. For ease of notation, define the (mini-max) objective

$$F(\alpha) \overset{\Delta}{=} \max_{\omega \in \Omega^{f_1}} f_1(\alpha, \omega).$$

Further, let **FSQP′-MOC**(α) denote the local minimizer obtained when Algorithm **FSQP′-MOC** is applied to problem (SS) with initial point α. The following algorithm statement is adapted from [4].

Algorithm MLSL

```
Step 0. set $\ell \leftarrow 1$, $\mathcal{M} \leftarrow \emptyset$.

Step 1. generate $N$ points $\alpha_{(\ell-1)N+1}, \ldots, \alpha_{\ell N}$ uniformly over $X$.
    set $i \leftarrow 1$.

Step 2. if ($\exists j$ s.t. $F(\alpha_j) < F(\alpha_i)$ and $\|\alpha_i - \alpha_j\|_s < r_\ell$) then goto Step 3.
    else set $\mathcal{M} \leftarrow \mathcal{M} \cup \{\textbf{FSQP}'\textbf{-MOC}(\alpha_i)\}$.

Step 3. set $i \leftarrow i + 1$.
    if $i \leq \ell N$ then goto Step 2.
    else set $\ell \leftarrow \ell + 1$ and goto Step 1.
```

It remains to specify how we select the critical distance r_ℓ, the definition of the metric $\|\cdot\|_s$ we use for signal sets (as parameterized by α), and how we generate the sample points. Following [4], we use

$$r_\ell = \frac{1}{\sqrt{\pi}} \left(\Gamma(1+n/2) \cdot m(X) \cdot \frac{\zeta \ln(\ell N)}{\ell N} \right)^{1/n},$$

where n is the number of variables (MK for our problem), $m(X)$ is the volume of the feasible region, and $\zeta > 2$. To compute $m(X)$, note that in view of symmetry with respect to the signals,

$$m(X) = A^M,$$

where A is the volume of the feasible region for the parameters corresponding to one signal (recall, M is the number of signals). The quantity A is easily estimated using a Monte Carlo technique.

Note that, for our problem, as far as the transmitter is concerned, a given signal set is unchanged if we were to swap the coefficients $\alpha^{m_1,k}$, $k = 1, \ldots, K$, with $\alpha^{m_2,1}$, $k = 1, \ldots, K$, where $m_1 \neq m_2$. The distance "metric" we use in Algorithm **MLSL** should take this symmetry into account. Consider the following procedure for computing the distance between signal sets parameterized by α_1 and α_2.

```
**function** dist($\alpha_1, \alpha_2$) {
    **for** $i = 1, \ldots, M-1$ **do** {
        $d_i = \min \left\{ \sum_{k=1}^{K} (\alpha_1^{i,k} - \alpha_2^{j,k})^2 \;\middle|\; j \in \{1, \ldots, M\} \setminus \{j_1, \ldots, j_{i-1}\} \right\}$
        $j_i = \arg\min \left\{ \sum_{k=1}^{K} (\alpha_1^{i,k} - \alpha_2^{j,k})^2 \;\middle|\; j \in \{1, \ldots, M\} \setminus \{j_1, \ldots, j_{i-1}\} \right\}$
    }
```

```
        **return** $\left(\sum_{i=1}^{M} d_i\right)^{1/2}$
}
```

This is not a metric in the strict sense of the definition, though it suffices for our purposes and we will set

$$\|\alpha_1 - \alpha_2\|_s \triangleq \operatorname{dist}(\alpha_1, \alpha_2).$$

To aid the generation of sample points, before starting the **MLSL** loop we compute the smallest box which contains the feasible set X. By symmetry with respect to the signals, we can do this by solving $2K$ linear programs. Specifically, let $\bar{\alpha}^k \in \mathbb{R}$, $k = 1, \dots, K$ be defined as the optimal value of the linear program (LP)

$$\begin{aligned} \max_{\alpha^{1,1},\dots,\alpha^{1,K}} \quad & \alpha^{1,k} \\ \text{s.t.} \quad & \sum_{q=1}^{K} \alpha^{1,q} \phi_k[t] \le C, \quad t = 1, \dots, T, \\ & \sum_{q=1}^{K} \alpha^{1,q} \phi_k[t] \ge -C, \quad t = 1, \dots, T, \end{aligned} \tag{U_k}$$

and let $\underline{\alpha}^k \in \mathbb{R}$, $k = 1, \dots, K$ be defined as the optimal value of the LP

$$\begin{aligned} \min_{\alpha^{1,1},\dots,\alpha^{1,K}} \quad & \alpha^{1,k} \\ \text{s.t.} \quad & \sum_{q=1}^{K} \alpha^{1,q} \phi_k[t] \le C, \quad t = 1, \dots, T, \\ & \sum_{q=1}^{K} \alpha^{1,q} \phi_k[t] \ge -C, \quad t = 1, \dots, T. \end{aligned} \tag{L_k}$$

Then, it should be clear that

$$X \subseteq \mathcal{B} \triangleq \{\ \alpha \in \mathbb{R}^{MK} \mid \alpha^{m,k} \in [\underline{\alpha}^k, \bar{\alpha}^k],\ k = 1, \dots, K,\ m = 1, \dots, M\ \}.$$

Using standard random number generators, it is a simple matter to choose samples from the uniform distribution on the box $\mathcal{B}$. Thus, for *Step 1* of Algorithm **MLSL**, we repeatedly generate samples from the uniform distribution on $\mathcal{B}$, discarding those which do not lie in X, until we find N which do lie in X. It should be clear that such a procedure is equivalent to drawing N samples from the uniform distribution on X.

7.4 Numerical Results

Following [15], we consider the noise distributions p_N listed in Table 7.1. For the definition of the Generalized Gaussian distribution, the constant a is defined as

$$a \triangleq \left(\frac{\sigma^2\Gamma(1/4)}{\Gamma(3/4)}\right)^{1/2}.$$

For our numerical experiments, we assume $\sigma = 1$. The case $K = 2$ is of common interest, and we use either a sine-sine basis

$$\left\{\sqrt{\frac{2}{T}}\sin(2\pi\omega_1 t/T),\ \sqrt{\frac{2}{T}}\sin(2\pi\omega_2 t/T)\right\},$$

or a sine-cosine basis

$$\left\{\sqrt{\frac{2}{T}}\sin(2\pi\omega_1 t/T),\ \sqrt{\frac{2}{T}}\cos(2\pi\omega_1 t/T)\right\},$$

where $\omega_1 = 10$ and $\omega_2 = 11$. When $K = 2$ we can display the results in the plane as familiar signal *constellations.* Finally, we run experiments for $M = 8,\ 16$ signals, $T = 50$ samples, and with an amplitude bound of $C = \sqrt{10}$. Note that, for $M = 16$, problem (SS) has 32 variables, 120 objective functions, and 800 constraints.

We ran Algorithm **MLSL** for 20 different problem instances. The algorithm was stopped after it appeared that no better local minimizers would be found

	$p_N(\tau)$	$K_N(\delta)$
Gaussian	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\tau^2}{2\sigma^2}\right)$	$\frac{\delta^2}{2\sigma^2}$
Laplacian	$\frac{1}{\sqrt{2\sigma^2}} \exp\left(\frac{-\sqrt{2} \cdot \lvert\tau\rvert}{\sigma}\right)$	$\frac{\sqrt{2} \cdot \lvert\delta\rvert}{\sigma} + \exp\left(\frac{-\sqrt{2} \cdot \lvert\delta\rvert}{\sigma}\right) - 1$
Hyperbolic Secant	$\frac{1}{2\sigma} \text{sech}\left(\frac{\pi\tau}{2\sigma}\right)$	$-2\ln\left(\text{sech}\left(\frac{\pi\delta}{4\sigma}\right)\right)$
Generalized Gaussian	$\frac{1}{2\Gamma(5/4)a} \exp\left(\frac{-\tau^4}{a^4}\right)$	$\frac{\Gamma^2(3/4)}{\Gamma^2(1/4)}\left(6\frac{\delta^2}{\sigma^2} + \frac{\delta^2}{\sigma^4}\right)$
Cauchy	$\frac{1}{\pi\sigma(1 + (\tau/\sigma)^2)}$	$\ln\left(1 + \frac{\delta^2}{4\sigma^2}\right)$

Table 7.1: Noise distributions and the associated KL distance function

(i.e. the estimate of the global minimum remained constant for several **MLSL** iterations). In Tables 7.2 and 7.3 we list our computed minimum values for instances of (SS) with $M = 8$ and $M = 16$, respectively. Note that our solutions agree with those reported in [15]. In all cases, execution was terminated after no more than 10 to 15 minutes. In Figures 7.2 through 7.7 we show the optimal signal constellations for several of the instances of (SS) corresponding to the optimal values listed in Tables 7.2 and 7.3.

In order to judge the efficiency of the RFSQP implementation, we compared its performance on the signal sets problem with two other widely available SQP codes. The first was VF02AD from the Harwell subroutine library [38], a standard SQP code based on Powell's algorithm [61]. As the code does not directly solve mini-max problems, we used the formulation suggested in [15] and solved

Noise	Basis	$F(\alpha^*)$
Gaussian	sine-sine	-69.793
	sine-cosine	-97.551
Laplacian	sine-sine	-63.122
	sine-cosine	-84.463
Hyperbolic Secant	sine-sine	-61.093
	sine-cosine	-83.196
Generalized Gaussian	sine-sine	-189.09
	sine-cosine	-264.18
Cauchy	sine-sine	-22.731
	sine-cosine	-30.673

Table 7.2: Optimal computed values for signal set design with $M = 8$

the problem

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^{MK},\gamma\in\mathbb{R}} \quad & -\gamma^2 - \epsilon_r\|\alpha\|_2^2 \\ \text{s.t.} \quad & f_1(\alpha,\omega) \le -\gamma^2, \quad \forall\omega\in\Omega^{f_1}, \\ & g_1(\alpha,\xi) \le 0, \quad \forall\xi\in\Xi^{g_1}, \\ & \gamma \ge 0, \end{aligned}$$

where ϵ_r, a “regularization” parameter, is small (possibly zero). In Table 7.4, we list the number of times VF02AD successfully converged to a local minimizer out of 20 trials for a given noise distribution and basis (and regularization parameter). For each trial the initial point was drawn from the uniform distribution over the feasible set. It is clear from the table that the standard SQP algorithm had little success converging to a local solution. The failures were essentially always due to inconsistent constraints in the QP sub-problem. As mentioned in

Noise	Basis	$F(\alpha^*)$
Gaussian	sine-sine	-29.314
	sine-cosine	-39.742
Laplacian	sine-sine	-32.370
	sine-cosine	-44.076
Hyperbolic Secant	sine-sine	-29.577
	sine-cosine	-40.500
Generalized Gaussian	sine-sine	-57.829
	sine-cosine	-76.138
Cauchy	sine-sine	-11.426
	sine-cosine	-15.688

Table 7.3: Optimal computed values for signal set design with $M = 16$

Noise	sine-sine	sine-cosine	sine-cosine
	$(\epsilon_r = 0)$	$(\epsilon_r = 0)$	$(\epsilon_r = 10^{-6})$
Gaussian	4	0	1
Laplacian	6	0	1
Hyperbolic Secant	5	0	0
Generalized Gaussian	6	0	0
Cauchy	2	0	0

Table 7.4: Number of successful runs for VF02AD out of 20 trials.

Section 7.3, this was a strong motivation for applying an algorithm generating feasible iterates to this problem. In our trials, RFSQP (as well as CFSQP) never failed to converge to, at least, a local solution.

In Table 7.5 we compare the average performance of RFSQP on the signal sets problem to that of CFSQP [36], an implementation of the feasible SQP algorithm due to Panier and Tits [51] (see also Section 2.4). For this table, we restricted our attention to the case of a sine-cosine basis and the generation of $M = 16$ signals. For each noise distribution, 10 initial points were drawn from the uniform distribution on the feasible set and both algorithms were run for each generated initial point. In the table we report the average number of iterations required to converge to a local solution, the average amount of time required, and the average amount of time per iteration. Averaging over the noise distributions, RFSQP took 74 iterations versus only 42 for CFSQP. This is to be expected, though, since RFSQP, an implementation of the algorithm **FSQP′-MOC** (see Chapter 5) uses an incomplete model at each iteration. On the other hand, RFSQP required only 39 seconds on average to converge to a local solution, versus 107 seconds for CFSQP. This clearly demonstrates the superiority of the algorithm **FSQP′-MOC** in cases where the time required to compute function evaluations does not dominate the time required to generate a new iterate. In these trials, RFSQP was approximately five times faster per iteration than CFSQP. As an aside, if it were the case that function evaluations were very expensive, then it would make more sense to use the **FSQP′-MM** algorithm in RFSQP. The performance would then be very similar to that reported for CFSQP.

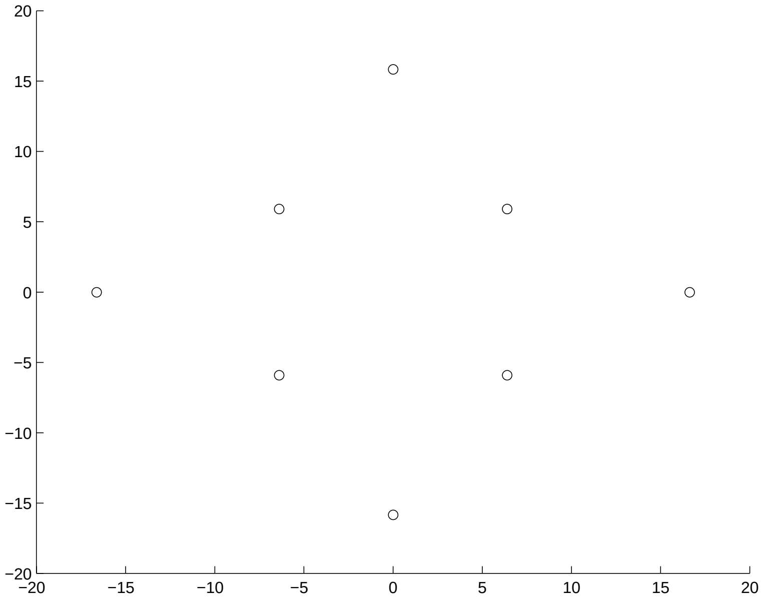

Figure 7.2: Optimal constellation for Gaussian noise, $M = 8$, sine-sine basis

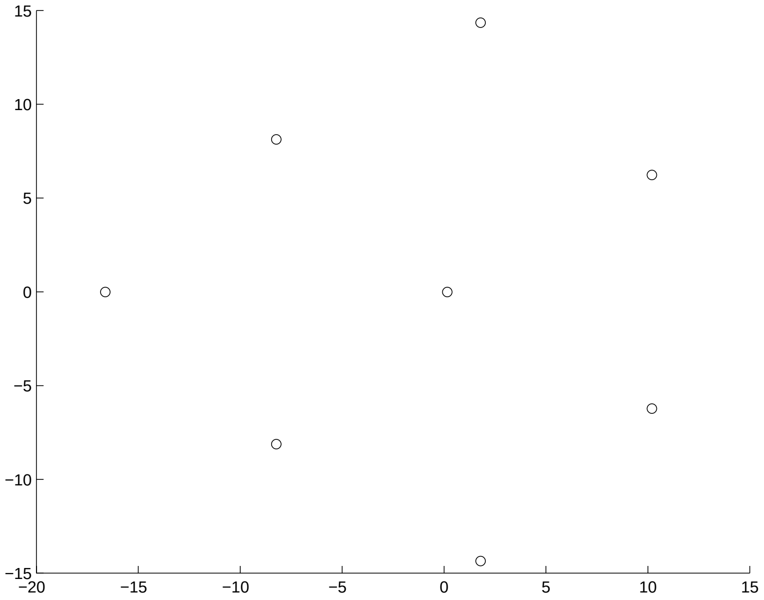

Figure 7.3: Optimal constellation for Generalized Gaussian noise, $M = 8$, sine-sine basis

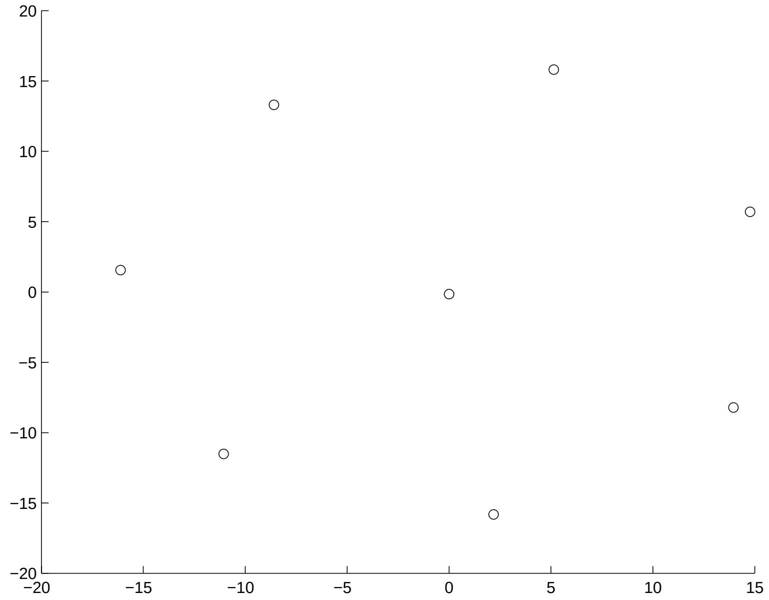

Figure 7.4: Optimal constellation for Laplacian noise, $M = 8$, sine-cosine basis

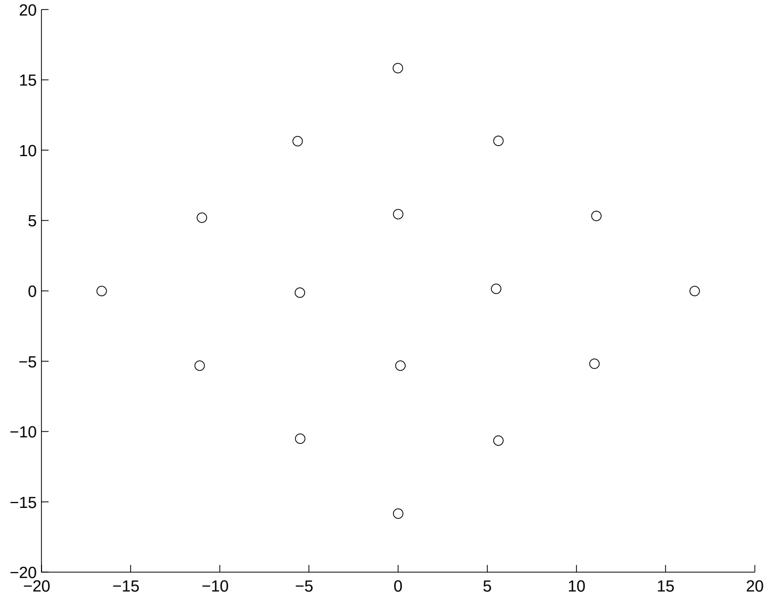

Figure 7.5: Optimal constellation for Cauchy noise, $M = 16$, sine-sine basis

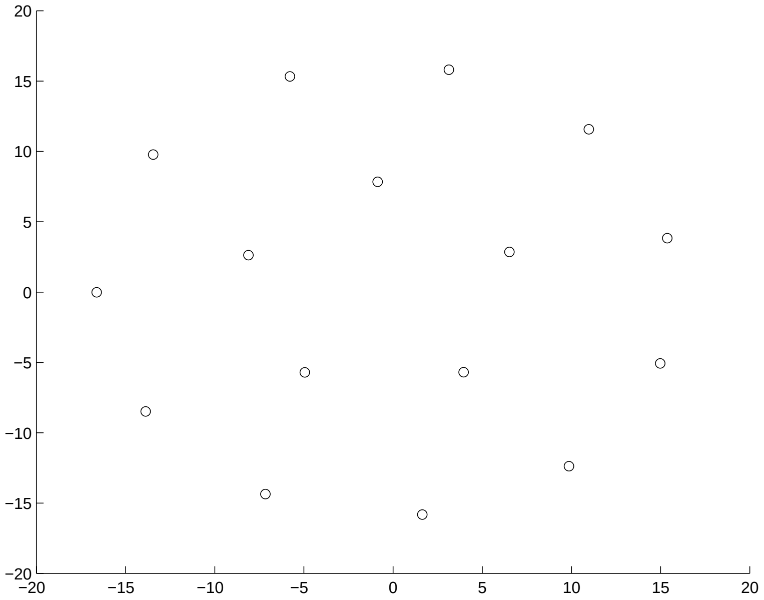

Figure 7.6: Optimal constellation for Cauchy noise, $M = 16$, sine-cosine basis

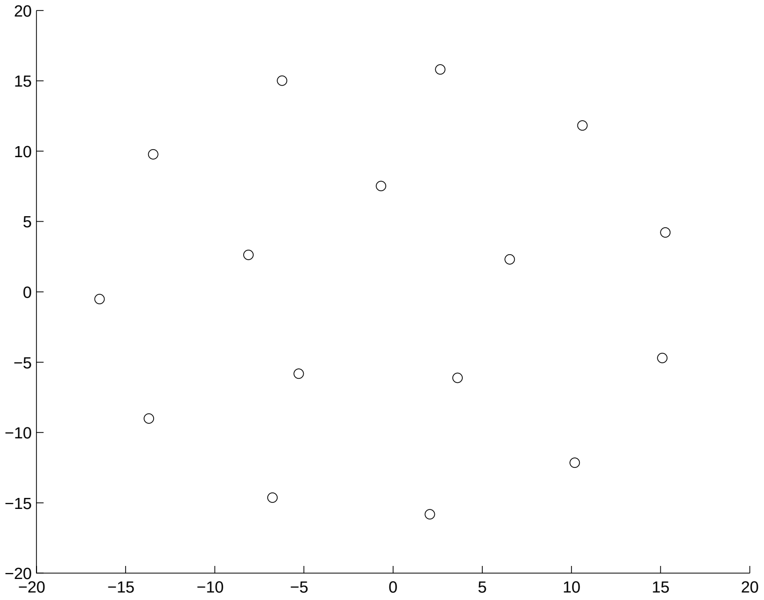

Figure 7.7: Optimal constellation for Hyperbolic Secant noise, $M = 16$, sine-cosine basis

		CFSQP	RFSQP
Gaussian	Iterations Time (sec) Time/Iteration	47 119 2.53	73 35 0.48
Laplacian	Iterations Time (sec) Time/Iteration	39 102 2.62	71 37 0.52
Hyperbolic Secant	Iterations Time (sec) Time/Iteration	37 98 2.65	75 43 0.57
Generalized Gaussian	Iterations Time (sec) Time/Iteration	36 84 2.33	78 40 0.51
Cauchy	Iterations Time (sec) Time/Iteration	52 133 2.56	73 38 0.52

Table 7.5: Average performance of RFSQP versus (non-SR) CFSQP.

Chapter 8

Conclusions

8.1 Overview

Motivated by problems from engineering analysis and design, we have developed a new SQP-type algorithm generating feasible iterates. The primary advantage of the algorithms presented in this dissertation is a dramatic reduction in the amount of computation required (over existing feasible SQP algorithms) in order to generate a new iterate. While this may not be very important for applications where function evaluations dominate the actual amount of work to compute a new iterate, it is very useful in many contexts. In any case, preliminary numerical results seem to indicate that decreasing the amount of computation per iteration did not come at the cost of increasing the number of function evaluations, and iterations, required to find a solution. It was shown that the basic algorithm is globally convergent and locally superlinearly convergent.

The basic algorithm was extended to handle problems with competing objective functions, i.e. the constrained mini-max problem. The mini-max structure was exploited in order to make the generation of a new iterate more efficient and maintain the objective descent properties of the basic algorithm. Again, the resultant algorithm was shown to be globally convergent and locally superlinearly convergent. The final extension involved incorporation of a scheme aimed at making the solution of problems with very many objectives and/or constraints more efficient. The idea was to carefully choose a small subset of "critical" objectives and constraints in order to construct the search direction at each iteration. The result was a dramatic reduction in the size of the QP sub-problems and number of gradient evaluations, without sacrificing any of the global and local convergence properties. The algorithms were all implemented in the portable ANSI C code RFSQP.

Finally, the implementation was used to solve a signal set design problem for detection in the presence of non-Gaussian noise. In that context, it was demonstrated that the algorithm performs very well as a local method embedded in a stochastic global optimization algorithm.

8.2 Future Work

A number of avenues exist for future work. To begin with, as with any optimization algorithm, the algorithms presented here are works in progress. Various extensions are possible, parameter tuning is necessary, and the implementation efficiency can always be improved. A few of the more important areas are listed below.

- It is possible to extend the class of problems (M) which are handled by the algorithm to include nonlinear equality constraints. Of course, we will not be able to generate feasible iterates for such constraints, but a scheme such as that studied in [34] could be used in order to guarantee asymptotic feasibility for equality constraints while maintaining feasibility for all inequality constraints.

- Using a method along the lines of those in [66], the algorithms of Chapter 5 could be used in an algorithm for directly tackling semi-infinite programming problems (without discretization).
- Work remains to be done to exploit the close relationship between the two least squares problems and the quadratic program as discussed in Section 6.4. A careful implementation should be able to use these relationships to great advantage computationally.
- More extensive testing and tuning of the algorithms should be done. Specifically, it would be useful to hook the implementation to the CUTE test set [6].

Appendix A

Code for the Application Example

Included in this appendix is the code used to evaluate the objective and constraint functions for the problem (SS) as given in Section 7.2. The code for the main program which calls RFSQP (and the implementation of the global algorithm) is not included, nor is any of the RFSQP code.

A.1 Main Header File

The following header file, signals.h, defines the main data structure for the problem and provides function prototypes for the utility functions given in Section A.3.

```
/****************************************************************/
/*  Main header file for optimal signal sets computation        */
/*                                                              */
/*  Craig Lawrence - June/July 1998                             */
/****************************************************************/

/* Includes */
#ifndef _MATH_H
#include <math.h>
#endif
#ifndef _STDIO_H
```

```
#include <stdio.h>
#endif
#ifndef _STDLIB_H
#include <stdlib.h>
#endif

/* Macros   */
#ifndef TRUE
#define TRUE 1
#define FALSE 0
#endif

/* Data structure defs */

enum basis_types {SIN_SIN, SIN_COS};
enum density_types {Gaussian, Laplacian, Hyperbolic_Secant,
                    Generalized_Gaussian, Cauchy};

struct SS_info_ {
   enum basis_types basis;
   enum density_types density;
   int K;             /* Number of basis functions */
   int M;             /* Number of signals to be designed */
   int N;             /* Number of time samples */
   double C;          /* Bound on signal amplitude */
   double sigma;      /* Standard deviation for density */
   double *frequ;     /* Basis function frequencies */
} *SS_info;

/* Function prototypes */
void   Initial_Alpha(double *, double *, double *);
double SS_basis(int, int);
double SS_signal(int, int, double *);
double SS_kldist(double);
double SS_klderiv(double);
```

A.2 Parameter Definition Header File

In this section, the header file which sets all of the algorithm parameters for RFSQP, `param.h`, is given in the form used for the signal sets design problem.

```
/*****************************************************************/
/*  CFSQPR1 - Algorithm parameter definition header             */
/*****************************************************************/

/* Tilting Parameter        */
#define ETA_0        1.e-2
#define C_0          1.e0
#define C_MIN        1.e-3
#define C_MAX        1.e3
#define C_FACTOR     1.e1

/* Second order correction dtilde  */
#define TAU          2.5
#define LS_DTILDE    1   /*  1 = use LS problem for dtilde
                             0 = use full QP for dtilde       */

/* Line Search             */
#define ALPHA        0.1e0
#define BETA         0.5e0

/* SR algorithm            */
#define DELTA_SR     1.e-6
#define EPSILON_SR   1.e0

/* Parameters for testing discretized SIP algorithm  */
#define USE_FULL     0   /*   1 = use all constr/obj at each
                                  iteration
                              0 = standard SIP algorithm
                                  (or eps-active)    */
#define EPS_ACT      0   /*   1 = use eps-act constr/obj at each
                                  iteration
                              0 = standard SIP algorithm
                                  (or full)          */
#define EPS_ACT_EPS  1.e-1  /* Tolerance for eps-active SIP
                                  approach           */

/* BFGS Update */
#define CHOLESKY     0   /*   1 = Maintain and update Cholesky
                                  factors of Hessian approx.  */

#define FSQP_TIME        /*   Keep track of time of execution */
```

A.3 Utility Functions

The following functions compute basis function values, signal values, KL distances, and derivatives of KL distances.

```
/****************************************************************/
/*  Utility functions for optimal signal sets computation       */
/*                                                              */
/*  Craig Lawrence - June - August 1998                         */
/****************************************************************/
#include "signals.h"

double sech(double);

double SS_basis(int j, int t)
{

  double f, pi, factor, dt;

  pi = 3.14159265358979e0;
  f = SS_info->frequ[j];
  factor = sqrt(2.e0/SS_info->N);
  dt = 1.e0/SS_info->N;

  switch (SS_info->basis) {
     case SIN_SIN:
        return sin(2.e0*pi*f*t*dt)*factor;
     case SIN_COS:
        if (j==0) return sin(2.e0*pi*f*t*dt)*factor;
        else return cos(2.e0*pi*f*t*dt)*factor;
  }

}

double SS_signal(int m, int t, double *x)
{
   int j;
   double signal=0.e0;

   for (j = m*SS_info->K; j < (m+1)*SS_info->K; ++j)
      signal += x[j]*SS_basis(j - m*SS_info->K, t);

   return signal;
```

```
}

double SS_kldist(double s)
{

   double sig = SS_info->sigma;

   switch (SS_info->density) {
      case Gaussian:
         return s*s/(2.e0*sig*sig);
      case Laplacian:
         return fabs(s)*sqrt(2.e0)/sig
                         + exp(-fabs(s)*sqrt(2.e0)/sig) - 1.e0;
      case Hyperbolic_Secant:
         return -2.e0*log(sech(atan(1.e0)*s/sig));
      case Generalized_Gaussian:
         return 0.11423664526112*(6.e0*s*s/(sig*sig)
                                  + s*s*s*s/(sig*sig*sig*sig));
      case Cauchy:
         return log(1.e0 + s*s/(4.e0*sig*sig));
   }
}

double SS_klderiv(double s)
{

   double sig = SS_info->sigma;
   double c1, d;

   switch (SS_info->density) {
      case Gaussian:
         return s/(sig*sig);
      case Laplacian:
         c1 = sqrt(2.e0)/sig;
         d = c1 - c1*exp(-fabs(s)*c1);
         if (s < 0) return -d;
         else if (s==0) return 0;
         else return d;
      case Hyperbolic_Secant:
         return 2.e0*atan(1.e0)*tanh(atan(1.e0)*s/sig)/sig;
      case Generalized_Gaussian:
         return 0.11423664526112*(12.e0*s/(sig*sig)
                                  + 4.e0*s*s*s/(sig*sig*sig*sig));
      case Cauchy:
         return 0.5e0*s/(sig*sig*(1.e0 + 0.25e0*s*s/(sig*sig)));
   }
}

double sech(double x)
{
   return 2.e0/(exp(x) + exp(-x));
}
```

A.4 Objective and Constraint Functions

Finally, in this section, we provide the functions which compute the actual objective and constraint values. These particular functions are written so that they may be called by RFSQP (see Section 6.1 for an explanation of the calling sequences).

```
/****************************************************************/
/* Objective and constraint evaluation functions for the        */
/* optimal signal set design problem (RFSQP format)             */
/*                                                              */
/* Problem posed as a true minimax problem with nonlinear       */
/* constraints                                                  */
/*                                                              */
/* Craig Lawrence  August, 1998                                 */
/****************************************************************/

#include "signals.h"

void
SS_obj(int nparam,int j,double *x,double *fj)
{
   int ind, m1, m2;
   double delta;

   j++;
   ind = j;
   m1 = 1; m2 = 2;
   while (ind > 1) {
      if (m2 >= SS_info->M) {
         m1++;
         m2 = m1 + 1;
      }
      else m2++;
      ind--;
   }
   m1--; m2--;

   *fj = 0.e0;
   for (ind=0; ind < SS_info->N; ind++) {
      delta = SS_signal(m1, ind, x) - SS_signal(m2, ind, x);
      *fj -= SS_kldist(delta);
   }
   return;
}

void
SS_grob(int nparam,int j,double *x,double *gradfj)
{
   int ind, m1, m2, t, k;
   double delta;

   j++;
   ind = j;
   m1 = 1; m2 = 2;
   while (ind > 1) {
      if (m2 >= SS_info->M) {
         m1++;
         m2 = m1 + 1;
      }
      else m2++;
      ind--;
   }
   m1--; m2--;

   for (ind=0; ind<SS_info->M*SS_info->K; ++ind)
      gradfj[ind] = 0.e0;

   for (t=0; t < SS_info->N; t++) {
      delta = SS_signal(m1, t, x) - SS_signal(m2, t, x);

      for (k=0; k<SS_info->K; ++k) {
         ind = m1*SS_info->K + k;
         gradfj[ind] -= SS_klderiv(delta)*SS_basis(k, t);
         ind = m2*SS_info->K + k;
         gradfj[ind] += SS_klderiv(delta)*SS_basis(k, t);
      }

   }
   return;
}

void
SS_cntr(int nparam,int j,double *x,double *gj)
{
   int t, m;
   double s;

   j++;
   t = (j - 1)%SS_info->N;
   m = (j - 1)/SS_info->N;
   s = SS_signal(m, t, x);
   *gj = s*s - SS_info->C*SS_info->C;

   return;
}

void
SS_grcn(int nparam,int j,double *x,double *gradgj)
{

   int t, m, l, k, ind;

   j++;
   t = (j - 1)%SS_info->N;
   m = (j - 1)/SS_info->N;

   for (l=0; l<SS_info->M; ++l) {
      for (k=0; k<SS_info->K; ++k) {
         ind = l*SS_info->K + k;
         if (l==m) gradgj[ind] =
                         2.e0*SS_basis(k, t)*SS_signal(m, t, x);
         else gradgj[ind] = 0.e0;
      }
   }

   return;
}
```

BIBLIOGRAPHY

[1] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, 1995.

[2] M. C. Biggs. Constrained minimization using recursive equality quadratic programming. In F. A. Lootsma, editor, *Numerical Methods for Non-Linear Optimization*, pages 411–428. Academic Press, New York, 1972.

[3] J. Birge, L. Qi, and Z. Wei. A variant of the Topkis-Veinott method for solving inequality constrained optimization problems. Technical Report AMR 97/29, School of Mathematics, The University of New South Wales, Sydney, Australia, 1997.

[4] C. G. E. Boender and H. E. Romeijn. Stochastic methods. In Reiner Horst and Panos Pardalos, editors, *Handbook of Global Optimization*, pages 829–869. Kluwer Academic Publishers, The Netherlands, 1995.

[5] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, pages 1–51, 1995.

[6] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 21:123–160, 1995.

[7] J. F. Bonnans, E. R. Panier, A. L. Tits, and J. L. Zhou. Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems – feasible iterates. *SIAM J. Numer. Anal.*, 29(4):1187–1202, August 1992.

[8] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. Technical Report 97-05, Optimization Technology Center, Argonne National Laboratory, 1997.

[9] R. M. Chamberlain, M. J. D. Powell, C. Lemaréchal, and H. C. Pedersen. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Programming Study*, 16:1–17, 1982.

[10] I. D. Coope and G. A. Watson. A projected Lagrangian algorithm for semi-infinite programming. *Math. Programming*, 32:337–356, 1985.

[11] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *J. Opt. Theory Appl.*, 89:507–541, 1996.

[12] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein. Optimization of two-dimensional signal constellations in the presence of Gaussian noise. *IEEE Transactions on Communications*, 22(1):28–38, 1974.

[13] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. Technical Report 97-4-08, Bell Laboratories, Computing Sciences Research Center, 1997.

[14] Philip E. Gill, Walter Murray, and Margaret Wright. *Practical Optimization.* Academic Press, New York, 1981.

[15] M. S. Gockenbach and A. J. Kearsley. Optimal signal sets for non-Gaussian detectors. *To Appear in SIAM J. Opt.*, 1998.

[16] D. Goldfarb. Factorized variable metric methods for unconstrained optimization. *Mathematics of Computation*, 30:796–811, 1976.

[17] Gene Golub and Charles Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[18] C. Gonzaga and E. Polak. On constraint dropping schemes and optimality functions for a class of outer approximation algorithms. *SIAM J. Control Optim.*, 17:477–493, 1979.

[19] C. Gonzaga, E. Polak, and R. Trahan. An improved algorithm for optimization problems with functional inequality constraints. *IEEE Transactions on Automatic Control*, AC-25:49–54, 1980.

[20] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.*, 23:707–716, 1986.

[21] S. A. Gustafson. A three-phase algorithm for semi-infinite programs. In A. V. Fiacco and K. O. Kortanek, editors, *Semi-Infinite Programming and Applications, Lecture Notes in Control and Information Sciences 215*, pages 138–157. Springer Verlag, 1983.

[22] S.-P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:453–473, 1977.

[23] S. P. Han. Superlinear convergence of a minimax method. Technical Report TR-78-336, Department of Computer Science, Cornell University, Ithaca, New York, 1978.

[24] S. P. Han. Variable metric methods for minimizing a class of nondifferentiable functions. *Math. Programming*, 20:1–13, 1981.

[25] J. N. Herskovits and L. A. V. Carvalho. A successive quadratic programming based feasible directions algorithm. In A. Bensoussan and J.L. Lions, editors, *Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986, Lecture Notes in Control and Information Sciences 83*, pages 93–101. Springer Verlag, Berlin, 1986.

[26] R. Hettich. An implementation of a discretization method for semi-infinite programming. *Math. Programming*, 34:354–361, 1986.

[27] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Rev.*, 35:380–429, 1993.

[28] W. Hock and K. Schittkowski. *Test Examples For Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems No. 187.* Springer-Verlag, Berlin, 1981.

[29] D. H. Johnson and G. C. Orsak. Relation of signal set choice to the performance of optimal non-Gaussian detectors. *IEEE Transactions on Communications*, 41(9):1319–1328, 1993.

[30] A. H. G. Rinooy Kan and G. T. Timmer. Stochastic global optimization methods; part II: multi-level methods. *Mathematical Programming*, 39:57–78, 1987.

[31] B. W. Kernighan and D. M. Ritchie. *The C Programming Language.* Prentice Hall, Englewood Cliffs, New Jersey, 2nd edition, 1988.

[32] K. C. Kiwiel. A phase I – phase II method for inequality constrained minimax problems. *Control and Cybernetics*, 12:55–75, 1983.

[33] K. C. Kiwiel. *Methods of Descent in Nondifferentiable Optimization, Lecture Notes in Mathematics No. 1183.* Springer-Verlag, Berlin, 1985.

[34] C. T. Lawrence and A. L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6:265–282, 1996.

[35] C. T. Lawrence and A. L. Tits. Feasible sequential quadratic programming for finely discretized problems from SIP. In R. Reemtsen and J.-J. Rückmann, editors, *Semi-infinite Programming*, pages 159–193. Kluwer Academic Publishers B.V., 1998. In the series Nonconvex Optimization and its Applications.

[36] C. T. Lawrence, J. L. Zhou, and A. L. Tits. *User's Guide for CFSQP Version 2.4: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, 1996. ISR TR-94-16r1, Institute for Systems Research, University of Maryland (College Park, MD).

[37] C. Lemaréchal. Nondifferentiable optimization. In G. Nemhauser, A. Rinooy-Kan, and M. Todd, editors, *Optimization, Handbooks in Operations Research and Management Science.* Elsevier Science, North Holland, 1989.

[38] Harwell Subroutine Library. *Library Reference Manual.* Harwell, England, 1985.

[39] David G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusets, 2nd edition, 1984.

[40] J. M. Maciejowski. *Multivariable Feedback Design.* Addison-Wesley, Wokingham, England, 1989.

[41] N. Maratos. *Exact Penalty Functions for Finite Dimensional and Control Optimization Problems.* PhD thesis, Imperial College of Science and Technology, 1978.

[42] D. Q. Mayne and E. Polak. A superlinearly convergent algorithm for constrained optimization problems. *Math. Programming Study*, 16:45–61, 1982.

[43] H. Mine, M. Fukushima, and Y. Tanaka. On the use of ϵ-most active constraints in an exact penalty function method for nonlinear optimization. *IEEE Transactions on Automatic Control*, AC-29:1040–1042, 1984.

[44] W. T. Nye and A. L. Tits. An application-oriented, optimization-based methodology for interactive design of engineering systems. *International J. of Control*, 43(6):1693–1721, 1986.

[45] K. Oettershagen. *Ein Superlinear Konvergenter Algorithmus zur Lösung Semi-Infiniter Optimierungsprobleme.* PhD thesis, Bonn University, 1982.

[46] J. M. Ortega and W. C. Rheinbolt. *Iterative Solution on Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[47] E. R. Panier and A. L. Tits. A superlinearly convergent method of feasible directions for optimization problems arising in the design of engineering systems. In A. Bensoussan and J.L. Lions, editors, *Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986, Lecture Notes in Control and Information Sciences 83*, pages 65–73. Springer Verlag, Berlin, 1986.

[48] E. R. Panier and A. L. Tits. A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. *SIAM Journal on Control and Optimization*, 25(4):934–950, 1987.

[49] E. R. Panier and A. L. Tits. A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design. *IEEE Transactions on Automatic Control*, AC-34(8):903–908, 1989.

[50] E. R. Panier and A. L. Tits. Avoiding the Maratos effect by means of a nonmonotone line search I. General constrained problems. *SIAM J. Numer. Anal.*, 28(4):1183–1195, August 1992.

[51] E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Math. Programming*, 59:261–276, 1993.

[52] E. R. Panier, A. L. Tits, and J. N. Herskovits. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM J. Control and Optimization*, 26(4):788–811, 1988.

[53] E. Polak. *Computational Methods in Optimization.* Academic Press, New York, 1971.

[54] E. Polak and L. He. Rate preserving discretization strategies for semi-infinite programming and optimal control. *SIAM J. Control and Optimization*, 30(3):548–572, 1992.

[55] E. Polak and D. Q. Mayne. An algorithm for optimization problems with functional inequality constraints. *IEEE Transactions on Automatic Control*, AC-21:184–193, 1976.

[56] E. Polak, D. Q. Mayne, and J. E. Higgins. A superlinearly convergent algorithm for min-max problems. In *Proceedings of the 28th Conference on Decision and Control*, December 1989.

[57] E. Polak, D. Q. Mayne, and J. E. Higgins. A superlinearly convergent algorithm for min-max problems. *Journal of Optimization Theory and Applications*, 69(3):407–439, 1991.

[58] E. Polak and A. L. Tits. A recursive quadratic programming algorithm for semi-infinite optimization problems. *Appl. Math. Optim.*, 8:325–349, 1982.

[59] H. V. Poor. *An Introduction to Signal Detection and Estimation.* Springer Verlag, New York, 1994.

[60] M. J. D. Powell. Convergence of variable metric methods for nonlinearly constrained optimization calculations. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 3*, pages 27–63. Academic Press, New York, 1978.

[61] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630*, pages 144–157. Springer Verlag, 1978.

[62] M. J. D. Powell. A tolerant algorithm for linearly constrained optimization calculations. *Math. Programming*, 45:547–566, 1989.

[63] J. G. Proakis. *Digital Communications.* McGraw Hill, New York, 1989.

[64] L. Qi and Z. Wei. Constant positive linear independence, KKT points, and convergence of feasible SQP methods. Technical Report AMR 97/4, School of Mathematics, The University of New South Wales, Sydney, Australia, 1997.

[65] R. Reemtsen. Discretization methods for the solution of semi-infinite programming problems. *J. Optim. Theory Appl.*, 71:85–103, 1991.

[66] R. Reemtsen and S. Görner. Numerical methods for semi-infinite programming: a survey. In R. Reemtsen and J.-J. Rückmann, editors, *Semi-infinite Programming*, pages 195–275. Kluwer Academic Publishers B.V., 1998. In the series Nonconvex Optimization and its Applications.

[67] R. Reemtsen and J.-J. Rückmann, editors. *Semi-infinite Programming.* Kluwer Academic Publishers B.V., 1998. In the series Nonconvex Optimization and its Applications.

[68] S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Math. Programming*, 7:1–16, 1974.

[69] B. Rustem and Q. Nguyen. An algorithm for the inequality-constrained discrete min-max problem. *SIAM J. Optimization*, 8(1):265–283, 1998.

[70] K. Schittkowski. *QLD: A Fortran Code for Quadratic Programming, User's Guide*. Mathematisches Institut, Universität Bayreuth, Germany, 1986.

[71] K. Schittkowski. Solving nonlinear programming problems with very many constraints. *Optimization*, 25:179–196, 1992.

[72] A. L. Tits, M. K. H. Fan, and E. R. Panier. Aspects of optimization-based CADCS. In *Proceedings of the IFAC Computer Aided Design in Control Systems Conference*, pages 47–57, 1988.

[73] T. Urban, A. L. Tits, and C. T. Lawrence. A primal-dual interior-point method for nonconvex optimization with multiple logarithmic barrier parameters and with strong convergence properties. Technical Report 98-27, Institute for Systems Research, University of Maryland, College Park, 1998.

[74] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, Princeton University, Dept. of Statistics and Operations Research, 1997.

[75] G. A. Watson. The minimax solution of an overdetermined system of nonlinear equations. *J. Inst. Math. Appl.*, 23:167–180, 1979.

[76] J. Zhou. *Fast, Globally Convergent Optimization Algorithms, with Applications to Engineering System Design*. PhD thesis, University of Maryland, Department of Electrical Engineering, 1992. ISR-TR Ph.D. 92-2.

[77] J. L. Zhou and A. L. Tits. Nonmonotone line search for minimax problems. *J. Optim. Theory Appl.*, 76:455–476, 1993.

[78] J. L. Zhou and A. L. Tits. An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions. *SIAM J. on Optimization*, pages 461–487, May 1996.

[79] J. L. Zhou, A. L. Tits, and C. T. Lawrence. *User's Guide for FSQP Version 3.7: A FORTRAN Code for Solving Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*, 1997. ISR TR-92-107r2, Institute for Systems Research, University of Maryland (College Park, MD).

[80] G. Zoutendijk. *Methods of Feasible Directions.* Elsevier Science, Amsterdam, The Netherlands, 1960.