2015

Practical Enhancements in Sequential Quadratic Optimization: Infeasibility Detection, Subproblem Solvers, and Penalty Parameter Updates

Hao Wang
Lehigh University

Practical Enhancements in Sequential Quadratic Optimization: Infeasibility Detection, Subproblem Solvers, and Penalty Parameter Updates

by

Hao Wang

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Doctor of Philosophy

in

Industrial Engineering

Lehigh University

May 2015

Approved and recommended for acceptance as a dissertation draft.

Date

Professor Frank E. Curtis
Dissertation Director

Accepted Date

Committee Members:

Professor Frank E. Curtis, Thesis Advisor

Professor Tamás Terlaky

Professor James V. Burke

Professor Katya Scheinberg

Acknowledgments

This thesis would not have been possible without the support of the professors, my fellow graduate students, and the staff in the Industrial and Systems Engineering Department of Lehigh University, as well as my family.

I would like to express my immense gratitude, first and foremost, to my academic advisor Professor Frank E. Curtis. I feel highly fortunate to have Professor Curtis as my advisor. His expertise and vision have guided me through my research, and played the roles of fuel and lighthouse in my exploratory journey of research. I also want to thank him for providing me with the opportunities to connect with great researchers through conferences and internship programs. Apart from research, I also received much sincere advice from him in many other aspects such as English writing skills and communication skills.

I would like to sincerely thank the remaining members of my thesis committee – Professor Tamás Terlaky, Professor James V. Burke, and Professor Katya Scheinberg – for sharing their knowledge and experience with me over the past few years. Professor Tamás Terlaky, the great leader of our department, shared with me many important insights, not only about my own research, but about topics such as Linear Optimization through the courses that he taught. Professor Katya Scheinberg has provided me with many valuable inspirational suggestions on my research. I also want to thank Professor James V. Burke, a co-author on two of our papers, for his invaluable input and stimulating discussions for several results research.

I want to convey my appreciation to the great and dedicated staff members in the ISE Department including Rita R. Frey, Kathy Rambo and many others. I sincerely thank these people whose dedicated work have made my life a lot easier.

During my six years study at Lehigh University, I am fortunate to have met friends such as Dan Li, Jiadong Wang, He Lin, Choat Inthawongse, Serdar Yildiz, Anahita Hassanzadeh, Xiaocun Que, Zheng Han, Yunfei Song and many others.

Finally, I would like to thank my family for their selfless love and support.

Contents

List of Figures

Abstract

The primary focus of this dissertation is the design, analysis, and implementation of numerical methods to enhance Sequential Quadratic Optimization (SQO) methods for solving nonlinear constrained optimization problems. These enhancements address issues that challenge the practical limitations of SQO methods.

The first part of this dissertation presents a penalty SQO algorithm for nonlinear constrained optimization. The method attains all of the strong global and fast local convergence guarantees of classical SQO methods, but has the important additional feature that fast local convergence is guaranteed when the algorithm is employed to solve infeasible instances. A two-phase strategy, carefully constructed parameter updates, and a line search are employed to promote such convergence. The first-phase subproblem determines the reduction that can be obtained in a local model of constraint violation. The second-phase subproblem seeks to minimize a local model of a penalty function. The solutions of both subproblems are then combined to form the search direction, in such a way that it yields a reduction in the local model of constraint violation that is proportional to the reduction attained in the first phase. The subproblem formulations and parameter updates ensure that near an optimal solution, the algorithm reduces to a classical SQO method for constrained optimization, and near an infeasible stationary point, the algorithm reduces to a (perturbed) SQO method for minimizing constraint violation. Global and local convergence guarantees for the algorithm are proved under reasonable assumptions and numerical results are presented for a large set of test problems.

In the second part of this dissertation, two matrix-free methods are presented for approximately solving exact penalty subproblems of large scale. The first approach is a novel

iterative re-weighting algorithm (IRWA), which iteratively minimizes quadratic models of relaxed subproblems while simultaneously updating a relaxation vector. The second approach recasts the subproblem into a linearly constrained nonsmooth optimization problem and then applies alternating direction augmented Lagrangian (ADAL) technology to solve it. The main computational costs of each algorithm are the repeated minimizations of convex quadratic functions, which can be performed matrix-free. Both algorithms are proved to be globally convergent under loose assumptions, and each requires at most $O(1/\varepsilon^2)$ iterations to reach ε-optimality of the objective function. Numerical experiments exhibit the ability of both algorithms to efficiently find inexact solutions. Moreover, in certain cases, IRWA is shown to be more reliable than ADAL.

In the final part of this dissertation, we focus on the design of the penalty parameter updating strategy in penalty SQO methods for solving large-scale nonlinear optimization problems. As the most computationally demanding aspect of such an approach is the computation of the search direction during each iteration, we consider the use of matrix-free methods for solving the direction-finding subproblems within SQP methods. This allows for the acceptance of inexact subproblem solutions, which can significantly reduce overall computational costs. In addition, such a method can be plagued by poor behavior of the global convergence mechanism, for which we consider the use of an exact penalty function. To confront this issue, we propose a dynamic penalty parameter updating strategy to be employed *within* the subproblem solver in such a way that the resulting search direction predicts progress toward both feasibility and optimality. We present our penalty parameter updating strategy and prove that does not decrease the penalty parameter unnecessarily in the neighborhood of points satisfying certain common assumptions. We also discuss two matrix-free subproblem solvers in which our updating strategy can be readily incorporated.

Chapter 1

Introduction

Mathematical constrained optimization has grown in recent decades to become one of the most important and influential tools in applied mathematics. However, the classes of nonlinear optimization (NLO) problems that can be solved analytically are extremely limited, especially due to issues such as potentially incompatible constraints, large problem size, nonconexity and degeneracy. As a result, computational optimization methods that can tackle those issues are critical for solving the complex problems arising in practice. In this dissertation we present techniques for addressing issues related to incompatible constraints and large scale problems.

Infeasibility detection, the first focus of this dissertation, is the process of reporting a valid certificate of infeasibility when given an infeasible optimization problem. Typically, such a certificate is given by a stationary point of constraint violation, thus named an *infeasible stationary point*. To rapidly detect infeasibility of the constraints is an important issue as many contemporary methods either fail or require an excessive number of iterations and/or function evaluations before being able to detect that a given problem instance is infeasible. As a result, modelers are forced to wait an unacceptable amount of time, only to be told eventually (if at all) that model and/or data inconsistencies are present. Rapid infeasibility detection is also important in techniques including branch-and-bound methods for nonlinear mixed-integer and parametric optimization, as algorithms for solving such problems often require the solution of a number of nonlinear subproblems. Slow infeasibility detection by such algorithms can create huge bottlenecks.

The second focus of this dissertation is the challenge of solving large-scale problems. In particular, we focus on solvers for NLO algorithm subproblems. Along with other various NLO methods, Sequential Quadratic Optimization (SQO), commonly known as SQP, has been a standard technique for solving constrained NLO problems for decades. With an appropriate globalization mechanism, SQO methods can converge from remote starting points. They are also revered for their fast local convergence guarantees and impressive practical performance. SQO methods also enjoy the characteristics of being able to be warm-started effectively and providing highly accurate solutions. However, the applicability of SQO has traditionally be limited to small-to-medium-scale problems. This is mainly because a sequence of inequality-constrained Quadratic Optimization (QO) subproblems of the same size as the origin problem need to be solved. This issue has been a persistent challenge in SQO when applied to large-scale problems. However, in many situations, an accurate solution to the subproblem of a SQO method is not necessarily needed. In fact, an approximate solution with a relatively accurate estimate of the active-set or yielding sufficient improvement toward optimality or feasibility is often adequate to ensure progress of the algorithm. This fact motivates us to craft methods that can rapidly find inexact solutions of SQO subproblems.

The acceptance of inexact subproblem solutions offers the possibility of terminating the subproblem solver early, perhaps well before an accurate solution has been computed. This characterizes the types of strategies that we focus on in the third part of this thesis. Recently, some work has been done to provide global convergence guarantees for SQP methods that allow inexact subproblem solves [27]. However, the practical efficiency of such an approach remains an open question. A critical aspect of any implementation of such an approach is the choice of subproblem solver. This is the case as the solver must be able to provide good inexact solutions quickly, as well as have the ability to compute highly accurate solutions—say, by exploiting well-chosen starting points—in the neighborhood of a solution of the NLO. In addition, while a global convergence mechanism such as a merit function or filter is necessary to guarantee convergence from remote starting points, an NLO algorithm can suffer when such a mechanism does not immediately guide the algorithm toward promising regions of the search space. To confront this issue when an exact penalty function is used as a merit function, we propose a dynamic penalty parameter updating strategy to be incorporated *within* the subproblem solver so that each computed search direction predicts progress toward both feasibility and optimality. This strategy represents a stark contrast to previously proposed techniques that only update the penalty parameter after a sequence of iterations [40] or at the expense of numerous subproblem solves within a single iteration [19, 13].

Overall, in this dissertation, we present algorithms for addressing issues related to infeasibility, the challenge of solving large scale problems, and complicating factors involved in updating penalty parameters. Specifically, this dissertation includes the following:

1. First, we develop, analyze, and discuss the implementation of a globally convergent SQO framework. Emphasis is placed on a solid theoretical foundation for its ability to rapidly converge to an infeasible stationary point in infeasible cases and an optimal solution in feasible cases. To our knowledge, this is a novel feature that most contemporary methods do not possess.

2. Second, we design, analyze, and compare two matrix-free algorithms for inexactly solving penalty QO subproblems in a generic penalty SQO framework. Both algorithms are able to rapidly find good inexact solutions. Our primary contribution is a new iterative-reweighting algorithm, for which we present a convergence proof and complexity analysis.

3. Finally, we introduce a basic penalty SQO algorithm that will form the framework for which we will introduce our penalty parameter updating strategy and matrix-free subproblem solvers. We discuss implementations of our methods and the results of extensive numerical experiments. Our main contribution is a novel technique for ren-

dering an appropriate value of the penalty parameter while solving the subproblem.

The structure of this dissertation is as follows. Chapter 2 discusses some background of nonlinear optimization methods, including the contemporary solvers and techniques related to the topics in this dissertation. In Chapter 3 we develop and analyze our proposed penalty-SQO method and investigate its global and local convergence behavior for both feasible and infeasible cases under common conditions. Chapter 4 presents two matrix-free solvers that can quickly find an inexact solution for subproblems with an exact penalty term. In Chapter 5, we propose an updating strategy for the penalty parameter while solving the SQO subproblem. Final remarks and comments on all of the methods in this dissertation are presented in Chapter 6.

Chapter 2

Background

2.1 Sequential Quadratic Optimization

We frame this dissertation in the context of the generic constrained NLO

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t.} \ & \begin{cases} c_{\mathcal{E}}(x) = 0 \\ c_{\mathcal{I}}(x) \leq 0 \end{cases} \end{aligned} \tag{2.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $c_{\mathcal{E}} : \mathbb{R}^n \to \mathbb{R}^{m_{\mathcal{E}}}$, and $c_{\mathcal{I}} : \mathbb{R}^n \to \mathbb{R}^{m_{\mathcal{I}}}$ are continuously differentiable. In this dissertation, we are particularly interested in problems where the constraints may be infeasible, or the number of variables n and the number of constraints $m := m_{\mathcal{E}} + m_{\mathcal{I}}$ are very large.

Among algorithms for solving NLO (2.1.1), SQO method has become one of most powerful methods. Ever since 1963, when it was first proposed by Wilson [82], SQO has evolved into a powerful class of methods for a wide range of constrained optimization problems. Define the Lagrangian to (2.1.1) as

$$\mathcal{L}(x, \lambda) := f(x) + \lambda_{\mathcal{E}}^T c_{\mathcal{E}}(x) + \lambda_{\mathcal{I}}^T c_{\mathcal{I}}(x),$$

where $\lambda = (\lambda_{\mathcal{E}}, \lambda_{\mathcal{I}})$ are the Lagrange multipliers. In a basic SQO approach, the search direction is defined as the solution to the following Quadratic Optimization (QO) subproblem, of which the objective function is a quadratic approximation to the Lagrangian at an iterate x^k, and the constraints are the linearizations of these in (2.1.1) at x^k:

$$\begin{aligned} &\min_{d} \ \nabla f(x^k)^T d + \tfrac{1}{2} d^T H^k d \\ &\text{s.t.} \ \begin{cases} c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T d = 0, \\ c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T d \leq 0. \end{cases} \end{aligned} \tag{2.1.2}$$

Here H^k is the exact or approximate Hessian of $\mathcal{L}(x, \lambda)$ at (x^k, λ^k) with respect to x.

SQO variants typically enjoy global convergence guarantees under certain common sets of assumptions when globalization techniques are employed. Early global convergence proofs were accomplished in [47, 68], which still provide the foundations for proving global convergence for many SQO methods. When explicit second-order derivative information is used, one can show that SQO methods behave like Newton's method to solve the Karush-Kuhn-Tucker (KKT) conditions for the NLO problem (2.1.1) including only the active constraints at the solution. This result is given by [71], which serves as the foundation for proving the local convergence rate for many SQO methods. SQO is famous for its fast local convergence in the neighborhood of a solution point satisfying common assumptions and an appropriate constraint qualification.

2.2 Contemporary Nonlinear Optimization Solvers

With the growing importance of optimization in many areas, researchers have implemented many successful optimization algorithms into off-the-shelf solvers, enabling people from different fields to conveniently apply the solvers for their own applications. Among many existing most successful NLO solvers, we focus on the software packages `Ipopt` [78], `Knitro` [16, 18, 80], and `Filter` [36]. All of them implement either an SQO or an Interior Point (IP) method, and are considered to be the leading computational tools for general-purpose

NLO problems. Next we review the methods implemented in these three solvers. Detailed descriptions of other solvers such as `Lancelot` [25], `Snopt` [40] and `Loqo` [50] are not included.

The basic structure of `Ipopt` is a primal-dual IP method framework, which solves a sequence of barrier problems with monotonically decreasing barrier parameters. For each value of the barrier parameter, the KKT system of the barrier problem is attacked by a damped Newton method to obtain the search direction. Then a step size is determined by evaluating the progress toward optimality along the search direction using a filter technique.

`Knitro` has three solvers for users to choose: `Knitro-Direct`, `Knitro-CG`, and `Knitro-Active`, with default option `Knitro-Direct`. `Knitro-Direct` implements the IP method proposed in [80], which solves a sequence of barrier problems to obtain an optimal solution. Upon computing the search direction for a given barrier problem, a backtracking line search is employed to determine the step size. The algorithm implemented in `Knitro-CG` is also an IP method, though it differs in that the search directions are obtained by (approximately) solving a QO subproblem with a trust region constraint. The final `Knitro` algorithm, `Knitro-Active`, is a sequential linear-quadratic programming algorithm. It first solves a Linear Optimization (LO) subproblem, which ends up with a set of "working" constraints (constraints are that satisfied as equalities) at the LO solution. This working set is used to formulate an equality-constrained QO subproblem. The search direction is a combination of the solutions of the LO subproblem and the QO subproblem.

`Filter` employs a SQO method, computing the search direction by solving a QO subproblem within a trust-region. A filter technique is used to decide whether the trial step should be accepted or rejected.

2.3 Infeasibility Detection in Contemporary Solvers

In theory, SQO methods can often guarantee global convergence from remote starting points to infeasible stationary points in addition to global and fast local convergence for feasible problems. For example, see the trust-region SQO method in [8] or the penalty line search SQO methods in [19, 21]. For those methods, it is shown that in the neighborhood of a solution point satisfying common assumptions, fast local convergence to feasible optimal solutions can be attained. However, most contemporary SQO methods make no attempt to achieve a theoretical fast rate of convergence to stationary points in the infeasible case. Rapid infeasibility detection can be shown by [13], though a major deficiency in that algorithm is that perhaps many QO subproblems are required to be solved in each iteration to drive fast local convergence.

For an algorithm to be robust and efficient in all practical situations, it should be able to return a certificate of infeasibility when given an infeasible instance of the NLO problem (2.1.1). Such a certificate can be developed, for example, through a measure of infeasibility of the constraints. For our purposes, we consider the following ℓ_1-norm constraint violation measure:

$$v(x) := \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} \max\{c_i(x), 0\}. \tag{2.3.1}$$

Other types of constraint violation measures can be obtained by choosing different norms; e.g., ℓ_2-norm-squared violation $v_2(x) := \sum_{i\in\mathcal{E}} (c_i(x))^2 + \sum_{i\in\mathcal{I}} (\max\{c_i(x), 0\})^2$ and the ℓ_∞-norm violation $v_\infty(x) = \max\{|c_i(x)|, i \in \mathcal{E}; \max\{c_i(x), 0\}, i \in \mathcal{I}\}$. Different measures may lead to distinct behavior when incorporated into an algorithm, but a discussion of these issues is outside the scope of this dissertation. If there is no point satisfying the constraints of problem (2.1.1), then algorithms should be designed to return a point minimizing constraint violation, i.e., in such cases, they should find the optimal solution to the following *infeasibility problem*:

$$\min_x \; v(x). \tag{2.3.2}$$

In practice, the priority for such an algorithm is to locate a stationary point for the NLO problem (2.1.1), but if that is deemed unattainable, then the algorithm should at least guarantee that a stationary point for the infeasibility problem (2.3.2) (i.e., a stationary point for v) will be found. A point that is locally stationary for the infeasibility problem

(2.3.2) but infeasible for NLO (2.1.1) is called an *infeasible stationary point* for (2.1.1), and the detection of such a point is a valid certificate of infeasible for problem (2.1.1). We are aware that finding an infeasible stationary point does not mean that the constraints of (2.1.1) are incompatible, but merely means they are locally inconsistent.

While many contemporary solvers have its infeasibility detection mechanism, in practice they may often fail to locate infeasible stationary points, or at least fail to do so in a timely manner. In some situations, this inefficiency may be due to the use of a "switching" technique to tackle potentially infeasible problems. The main intent of such algorithms is to attempt to solve the NLO problem (2.1.1) until it has been determined that further progress cannot be obtained, at which point they revert to solving the infeasibility problem (2.3.2) directly. That is, they revert to attempting to improve constraint satisfaction during a *feasibility restoration phase*. If sufficient progress in minimizing constraint violation is attained in the restoration phase, then the method returns to the main algorithm; otherwise, an infeasible stationary point may be detected. Such switching approaches are often effective, but they may lead to inefficiencies for infeasible cases or even certain types of feasible problems. The main reason for this is that, during the feasibility restoration phase, an algorithm may obtain a reduction in constraint violation, but at the same time it may be moving away from the set of optimal solutions. (Specifically, by ignoring the objective function completely during the feasibility restoration phase, the method may impair its overall progress.) If such an occurrence happens many times, the cost may be numerous iterations before escaping this cycle and moving to a new area or claiming infeasibility. Another inherent difficulty with a switching approach is that it is not easy to determine when to make a switch between the two phases. All of this motivates us in Chapter 3 to craft methods that can balance the two tasks of attaining optimality and constraint violation minimization without the use of a switching technique.

Another method for infeasibility detection involves the use of a penalty function. If the associated penalty parameter tends to an extreme value, then the algorithm transitions to minimizing constraint violation. We believe that this is a reasonable approach, but may also be inefficient if the convergence of the parameter to its extreme value occurs too slowly.

In the remainder of this subsection, we illustrate performance of the infeasibility detection techniques implemented in `Ipopt`, `Knitro` and `Filter`. In `Ipopt`, the restoration phase is triggered whenever the KKT system is ill-conditioned or the line search procedure ends up with a tiny step size. In the restoration phase, it improves constraint satisfaction by solving an ℓ_2-norm feasibility problem with an additional proximal term in the objective which prevents the new iterate from straying far from the previous iterate. This is motivated by the concern that the restoration phase may impair overall progress if left unchecked. Infeasibility is detected when an insufficient improvement in the restoration phase is obtained. Otherwise, it reverts to the optimization phase with a point that is closer to feasibility.

As in `Ipopt`, `Knitro-Direct` handles infeasibility detection via a condition on the step size. If it is too small, then the algorithm reverts to `Knitro-CG`. Infeasibility detection in `Knitro-CG` is handled through updates of a penalty parameter, which may tend to infinity in order to place an increasingly higher priority on minimizing constraint violation. In `Knitro-Active`, infeasibility is detected whenever the penalty parameter tends to infinity and the minimizer of an ℓ_1-norm model of the linearized constraints does not lead to an improvement in feasibility.

In `Filter`, when an infeasible QO is encountered, the algorithm turns to a feasibility restoration phase. Before entering the restoration phase, their QO solver, an active set method, returns a solution with some linear constraints being feasible and some infeasible. Consider the case where only inequalities exist in (2.1.1). In the restoration phase, the constraint index set $\mathcal{I}$ is partitioned into two sets, call them $\mathcal{I}_1(x^k)$ and $\mathcal{I}_2(x^k)$. The first set $\mathcal{I}_1(x^k)$ contains all the linear feasible constraints, and the other, $\mathcal{I}_2(x^k)$, contains the remaining indices, i.e., the linear infeasible constraints. While keeping the feasible linear constraints feasible, they minimize the ℓ_1 violation of the linear infeasible constraints. In the restoration phase, the feasibility restoration phase algorithm is basically an SQO method applied to such a problem. The restoration phase is exited whenever an iterate is determine at which the linearized constraints are all feasible. Otherwise, the algorithm, if it eventually detects infeasibility, returns a stationary point to the minimization of the violation of constraints in $\mathcal{I}_1(x^k)$, while satisfying the remaining constraints. Therefore, this type of infeasible stationary point is defined as one being stationary for the partial constraint violation measure

$$\sum_{i \in \mathcal{I}_1(x^k)} \max\{c^i(x), 0\} \tag{2.3.3}$$

subject to the constraints $c^i(x) \leq 0, i \in \mathcal{I}_2(x^k)$.

Overall, the solvers described above use heuristics to handle infeasibility, mostly applying a switching technique of the type previously mentioned. To show the performance of their strategies in practice, we provide a few small-scale examples (each with only two or three variables), all of which are infeasible and have the infeasible stationary point located at the origin. Typically, algorithms for NLO require a few assumptions to guarantee nice convergence properties: *regularity*, which means the gradients of the equality constraints and active inequality constraints are linearly independent; *strict complementarity*, which requires the Lagrange multipliers for the equality constraints and active inequality constraints to be nonzero; *second-order sufficiency*, which implies the Hessian of the Lagrangian function is (sufficiently) positive definite on the null space of gradients of the equality constraints and active inequality constraints. In order to have a variety of interesting test problems, the ones we have constructed satisfy different combinations of these assumptions (when observed in the context of the infeasibility problem (2.3.2) after slack variables are added to produce a constrained problem); in total we end up with eight different combinations which are listed in the Table 1 (Y=Yes, N=No). For Example 4 and Example 6 where the regularity condition does not hold, we observe that the multipliers cannot be uniquely determined, and some of them violate strictly complementarity. Therefore, it is not possible to create an instance with the regularity condition violated and strictly complementarity always satisfied, so we indicate Y/N for strict complementarity for such cases. The formulations of these examples are given in Appendix 1. Another problem, Example 9 — which is not mentioned in the table — is a feasible problem for which we find some curious results from `Filter`.

If the infeasibility of these examples is known a priori, then the user can directly apply the solvers to minimize the constraint violation. Table 2.1 provides these test results, where Iter. denotes the number of iterations and Eval. denotes the function evaluation counts. These solvers mostly require 20-30 iterations and function evaluations. Now suppose the infeasibility of these examples is not known by the user, so the solvers are tasked to solve the given NLO problems as they are stated in the Appendix. The performance information for this case without any presolve phase is provided in Table 2.2, where an asterisk means either the maximum iterations are exceeded or wrong information (e.g., that the problem is feasible) is reported to the user. The results overall seem unimpressive and far from the results in Table 2.2. We did not include the results of `Knitro-CG`, `Lancelot`, `Snopt`, and `Loqo` on these examples in the table since they always ran out of iterations without detecting infeasibility. The one solver that does produce acceptable-looking results is `Filter`, it reports correct information rapidly but fails to provide an accurate infeasible stationary point. For Example 9, `Filter` reports the problem is locally infeasible at the point $(-2.4109, -0.1012)$, but this is not actually an infeasible stationary point.

For cases where function evaluations are expensive or the problem scale is large, these solvers may run for hours or days without producing useful results, or, in the case of `Filter`, it may correct infeasible quickly, but incorrectly on certain problems. Therefore, one can see that nearly all these solvers may struggle for detecting infeasibility in practice.

2.4 QO Subproblems in Contemporary Solvers

There may be numerous issues arising when using subproblem (2.1.2). First, the constraints may be inconsistent, which makes such a basic SQO subproblem not well-defined in practice. In the past decades, various ingredients have been proposed to enhance SQO methods to avoid this difficulty. Generally, many of them focus on two aspects: how to

Table 2.1: Performance measures for solving the feasibility problem.

Problem	Ipopt		Knitro-Direct		Knitro-Active		Filter	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	28	29	14	15	13	24	17	21
2	31	32	31	33	8	9	12	13
3	50	131	10	11	9	12	12	13
4	24	79	18	29	10	13	10	12
5	166	786	29	40	21	24	30	32
6	37	48	20	21	19	22	26	27
7	59	65	31	34	19	20	25	28
8	46	71	19	20	23	28	26	29
9	28	29	18	19	18	54	15	14

Table 2.2: Performance measures for solving the NLO problem.

Problem	Ipopt		Knitro-Direct		Knitro-Active		Filter	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	48	281	38	135	22	235	16	16
2	109	170	*10000	*10000	23	167	12	12
3	788	3129	12	83	9	202	10	10
4	46	105	25	61	10	201	11	11
5	72	266	*1060	*3401	18	45	26	26
6	63	141	*76	*264	16	37	27	27
7	87	152	*10000	*43652	*10000	*20091	30	30
8	104	206	33	97	41	560	28	28
9	60	135	30	33	16	31	*13	*2

determine a step, and how to evaluate the improvement made by a step. For the former aspect, modern SQO methods involve varying techniques, typically depending on whether the method is based on a line search or trust-region approach. Both of them need some "mechanism" to evaluate the progress made by a step, of which there are commonly two kinds: a filter techinique or a penalty function formed by

$$\phi(x;\rho) = \rho f(x) + v(x).$$

By blending different options for all of these aspects, researchers have developed different versions of SQO, such as penalty SQO methods with line search (see [19] for example), penalty trust region SQO methods (see [35] for example), filter SQO methods with line search (see [79] for example), and filter trust region SQO methods (see [37] for example).

Each of these techniques yields different practical behavior.

Various other techniques are also proposed to handle other specific issues in SQO methods. The method of Sℓ_1QP [35], also known as the elastic SQO method [4, 40] was proposed to overcome the difficulties caused by the inconsistency of the subproblem (2.1.2) constraints. The subproblem in such a penalty SQO method is given by

$$\begin{aligned} \min_{d,r,s,t} \; & \rho \nabla f(x^k)^T d + \tfrac{1}{2} d^T H^k d + e^T(r+s) + e^T t \\ \text{s.t.} \; & \begin{cases} c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T d = r - s, \; i \in \mathcal{E} \\ c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T d \leq t, \; i \in \mathcal{I} \\ r \geq 0, s \geq 0, t \geq 0 \\ \|d\|_\infty \leq \Delta_k \end{cases} \end{aligned} \tag{2.4.1}$$

where constant $\rho > 0$ is penalty parameter. This subproblem is always feasible, meaning that the search direction is always well-defined.

Clearly, the major computational cost in an SQO method is the solution of the QO subproblems, which generally involve equality and inequality constraints. On one hand, near the optimal solution, once the active set is determined, the QO subproblems then reverts to equality constrained QOs, which can be solved by solving a system of linear equations. As a result, fast local convergence can be achieved when active set method is employed as the subproblem solver. This feature has led to active set SQO methods being considered powerful solvers for small-to-medium-scale NLOs.

On the other hand, the need of having to solve QO subproblems has prevented SQO methods from becoming an ideal option for large-scale problems. The existence of inequalities generally makes the QO subproblems difficult to solve. When active set QO solvers are used, the iterations needed for subproblems could be exponential with the number of constraints in the worst case. Consequently, the computational time grows rapidly when the problem size gets larger, and the method could even fail for large-scale cases in a timely manner.

An illustration of such an effect can be found in the COPS test set report [29]. We use an example from [29] to show how the performance of an SQO solver could suffer from large problem sizes. The comparison is carried out on `Filter`, `Knitro`, `Loqo`, `Minos`, and `Snopt`. This example is to find the polygon of maximal area, among polygons with n_v sides and diameter $d \leq 1$. The problem size is summarized in Table 2.3. The test results are shown in Table 2.4, where the first row for each solver is the computational time, f, "c violation" and "optimality" in the table represent the function value, constraint violation, and the optimality error at the final iterate, respectively. A † mark means an incorrect result was obtained, and ‡ represents a failure within time limit. One can see from Table 2.4 that the computational time grows dramatically for `Filter`, `Loqo`, `Minos`, and `Snopt` when the problem size becomes larger. They fail to solve the problem within the time limit, or even return an incorrect result (`Minos` with $n_v = 200$). Another aspect to notice is that the dramatic growth in the computational time of `Filter`, which may be due to the fact that an active set QO method is employed in this solver [36]. Among all the NLO solvers, `Knitro` is reported to be most stable, with slow growth in computation time. This is mainly due to the fact that all the three methods implemented in `Knitro` solve subproblems of linear equations, linear optimization, and equality constrained QO which is equivalent to linear equations. The same effect can be observed in most of the other examples in [29]. Therefore, the performance of SQO solvers on large-scale problems relies critically on the efficiency of the QO solvers.

Table 2.3: Largest-small polygon problem data

Variables	$2(n_v - 1)$
Constraints	$(\frac{1}{2}n_v + 1)(n_v - 1) - 1$
Bounds	$2(n_v - 1)$
Linear equality constraints	0
Linear inequality constraints	$n_v - 2$
Nonlinear equality constraints	0
Nonlinear inequality constraints	$\frac{1}{2}n_v(n_v - 1)$
Nonzeros in $\nabla^2 f(x)$	$11(n_v - 1) - 8$
Nonzeros in $c'(x)$	$2n_v(n_v - 1) - 2$

Finally, we comment that in many cases an exact solution of the QO subproblems are

Table 2.4: Performance on largest small polygon problem

Sover	$n_v = 50$	$n_v = 100$	$n_v = 200$
`Filter`	27.64s	555.2s	‡
f	7.66131e-01	7.77239e-01	‡
c violation	8.88e-16	1.17e-14	‡
optimality	8.96e-07	9.90e-07	‡
`Knitro`	1.41s	8.99s	59.53s
f	7.60725e-01	7.37119e-01	6.740980e-01
c violation	0.00e+00	0.00e+00	0.00e+00
optimality	7.53e-07	3.99e-07	2.01e-07
`Loqo`	14.39s	‡	‡
f	7.63694e-01	‡	‡
c violation	1.08e-10	‡	‡
optimality	1.02e-10	‡	‡
`Minos`	5.6s	121.3s	223.94s
f	7.66297e-01	6.79085e-01	6.57163e-01†
c violation	8.03-14	1.75e-13	2.66e-15†
optimality	6.32e-08	9.50e-10	9.55e-02†
`Snopt`	4.34s	69.35s	‡
f	7.84015e-01	7.85023e-01	‡
c violation	1.11e-10	1.78e-11	‡
optimality	8.30e-07	1.35e-07	‡

not necessary. If a given inexact setp can make sufficient progress toward a solution of the NLO, then global convergence can still be guaranteed. A variety of SQO methods with inexactness in step computations have been proposed recently [52, 58, 48]. The conditions that guarantee the global convergence of inexact SQP steps are discussed in [20]. This feature of the QO subproblem provides the foundation of applying SQO methods to large-scale applications.

2.5 Penalty Parameter Update in Contemporary Methods

The ℓ_1 penalty function $\phi(x;\rho)$ has been proved to be an *exact* penalty function of the NLO problem (2.1.1). This result is given by the following theorem [84, Theorem 17.3]

Theorem 2.5.1. *Suppose that x^* is a strict local solution of the nonlinear optimization problem* (2.1.1)*, at which the first order stationary conditions are satisfied, with Lagrange* *multipliers λ^*. Then x^* is a local minimizer of $\phi(x;\rho)$ for all $\rho < \rho^*$, where $\frac{1}{\rho^*} = \|\lambda^*\|_\infty$. If, in addition, the second-order sufficient conditions hold and $\rho < \rho^*$ then x^* is a strict local minimizer of $\phi(x;\rho)$.*

Exact penalty methods, such as Sℓ_1QP, solve a local approximation model of $\phi(x;\rho)$ such as

$$\min_{\|d\|_\infty \in \Delta_k} \rho \nabla f(x^k)^T d + \tfrac{1}{2} d^T H^k d + \sum_{i\in\mathcal{E}} |c_i(x^k) + \nabla c_i(x^k)^T d| + \sum_{i\in\mathcal{I}} \max\{c_i(x^k) + \nabla c_i(x^k)^T d, 0\},$$

which is equivalent to subproblem (2.4.1). Therefore, exact penalty methods are well as effective techniques for handling the inconsistent constraints arising in the QO subproblem.

Based on Theorem 2.5.1, the penalty parameter ρ is updated at each iteration to prevent the convergence to an undesirable point. Despite advantages of exact penalty methods, their performance can be significantly affected by the penalty parameter, making it difficult to keep efficient over a wide range of problems. Therefore, the penalty parameter updating strategy is the critical point of constructing penalty methods. It has proved difficult to designing successful updating strategy.

Early updating strategies may solve the subproblem for a finite sequence of decreasing ρ [40], or scale down the penalty parameter by the magnitude of the multipliers [34]. This strategy can result in inefficient behavior and also requires heuristics to terminate the update. Various approaches have been recently proposed to handle this situation. In [21], the penalty parameter ρ is updated at every iteration so that sufficient progress toward feasibility and optimality is guaranteed to first order. They solve an auxiliary LO subproblem to evaluate the progress. *Steering rules* [19] and other methods [13] also require the penalty parameter to guarantee sufficient improvement on feasibility and optimality. Multiple subproblems may be solved per iteration with decreasing penalty parameter until the desirable value is found.

Chapter 3

An SQO Algorithm with Rapid Infeasibility Detection

3.1 Introduction

Sequential quadratic optimization (SQO) methods are known to be extremely efficient when applied to solve nonlinear constrained optimization problems [47, 67, 82]. Indeed, it has long been known [7, 8, 9] that with an appropriate globalization mechanism, SQO methods can guarantee global convergence from remote starting points to *feasible* optimal solutions, or to *infeasible* stationary points if the constraints are incompatible. One of the main additional strengths of SQO is that in the neighborhood of a solution point satisfying common assumptions and an appropriate constraint qualification, fast local convergence to *feasible* optimal solutions can be attained [70].

Despite these important and well-known properties of SQO methods, there is an important feature that many contemporary SQO methods lack, and it is for this reason that the algorithm in this chapter has been designed, analyzed, and tested. Specifically, in addition to possessing the convergence guarantees mentioned in the previous paragraph, we have proved that the algorithm proposed in this chapter yields *fast local convergence when applied to solve infeasible problem instances.*

There are two main novel features of our algorithm. Most importantly, it is an algorithm that possesses global and local superlinear convergence guarantees for feasible *and* infeasible problems *without* having to resort to feasibility restoration. This feature, in that a single approach is employed for solving both feasible and infeasible problems, means that the algorithm avoids many of the inefficiencies that may arise when contemporary methods are employed to solve problems with incompatible constraints. The second novel feature of our algorithm is that it is able to attain these strong convergence properties with at most two quadratic optimization (QO) subproblem solves per iteration. This is in contrast to recently proposed methods that provide rapid infeasibility detection, but only at a much higher per-iteration cost.

In the following section, we compare and contrast our approach with recently proposed SQO methods, focusing on properties of those methods related to infeasibility detection. We then present our algorithm in §3.2 and analyze its global and local convergence properties in §3.3. Our numerical experiments in §5.6 illustrate that an implementation of our algorithm yields solid results when applied to a large set of test problems.

We remark at the outset that we analyze the local convergence properties of our algorithm under assumptions that are classically common for analyzing that of SQO methods. We explain that our algorithm can be backed by similarly strong convergence guarantees under more general settings (see our discussion in §3.3.3), but have made the conscience decision to use these common assumptions to avoid unnecessary distractions in the analysis. Overall, the main purpose of this chapter is to focus on the novelties of our algorithm—which include the unique formulations of our subproblems, our use of separate multiplier estimates for the optimization and a corresponding feasibility problem, and our unique combination of updates for the penalty parameter—which provide our algorithm with global and fast local convergence guarantees on both feasible and infeasible problem instances.

3.1.1 Literature Review

Our algorithm is designed to act as an SQO method for solving an optimization problem when the problem is feasible, and otherwise it is designed to act as a perturbed SQO method [26] for a problem to minimize constraint violation. In this respect, our method has features in common with those in the class of penalty-SQO methods [35] where search directions are computed by minimizing a quadratic model of the objective combined with a penalty on the violation of the linearized constraints. In such algorithms, if the penalty parameter is driven to an extreme value, then the algorithm transitions to solely minimizing constraint violation. We believe that this approach is reasonable, though there are two main disadvantages of the manner in which penalty-SQO methods are often implemented. One disadvantage is that the penalty parameter takes on all of the responsibility for driving constraint violation minimization. This leads to a common criticism of penalty methods, which is that the performance of the algorithm is too highly dependent on the penalty parameter updating scheme. The second disadvantage is that, if the penalty parameter is not driven to its extreme value sufficiently quickly, then convergence, especially for infeasible problems, can be slow. These disadvantages motivate us to design a method that reduces to a classical SQO approach for feasible problems, and where updates for the penalty parameter lead to rapid convergence in infeasible cases.

The immediate predecessor of our work is the penalty-SQO method proposed in [13]. In particular, the approach in [13] is also proved to yield fast local convergence guarantees for infeasible problems. That method does, however, have certain practical disadvantages. The most significant of these is that, particularly in infeasible cases, the method may require the solution of numerous QO subproblems per iteration. Indeed, near an infeasible stationary point, at least three QO subproblems must be solved. The first will reveal that for the current penalty parameter value it is not possible to compute a linearly feasible step, the second then gauges the progress toward linearized feasibility that can be made locally, and the third may produce the actual search direction. (In fact, if the conditions necessary for global convergence are not satisfied after the third QO subproblem solve, then even more QO subproblem solves are needed until the conditions are satisfied.) In contrast, the algorithm proposed in this chapter solves *at most two* QO subproblems per iteration. It also relies less on the penalty parameter for driving constraint violation minimization, and involves separate multiplier estimates for the optimization and feasibility problems. This last feature of our algorithm—that of having two separate multiplier estimates—is quite unique for an optimization algorithm. However, we believe that it is natural as the optimization algorithm must implicitly decide which of two problems to solve: the given optimization problem or a problem to minimize constraint violation.

Our algorithm is a multi-phase active-set method that has similarities with other such methods that have been proposed over the last few decades. For instance, the method in [13] borrows the idea, proposed in [19] and later incorporated into the line-search method in [17], of "steering" the algorithm with the penalty parameter. Consequently, that method at least suffers from the same disadvantages as the method in [13] when it comes to infeasibility detection. More commonly, multi-phase SQO methods have taken the approach of solving a first-phase inequality-constrained subproblem—typically a linear optimization (LO) subproblem—to estimate an optimal active set, and then solving a second-phase equality-constrained subproblem to promote fast convergence; e.g., see [21, 15, 23, 31, 32, 39]. A method of this type that solves two QO subproblems is that in [57], though again the second-phase subproblem in that method is equality-constrained as it only involves linearizations of constraints predicted to be active at an optimal solution. Our algorithm differs from these in that we do no active-set prediction, and rather solve up to two inequality-constrained subproblems. The methods in [44, 45] involve the solution of up to three subproblems per iteration: one to compute a "predictor" step, one to compute a "Cauchy" step, and one to compute an "accelerator" step. In fact, various subproblems are proposed for the "accelerator" step, including both equality-constrained and inequality-constrained alternatives. Our algorithm differs from these in that ours is a line search method, whereas they are trust region methods, and our first-phase subproblem computes a pure feasibility step rather than one influenced by a local model of the objective. This latter feature makes our method similar to those in [7, 8], though again our work is unique in that we ensure rapid infeasibility detection, which is not provided by any of the aforementioned methods besides that in [13]. Finally, we mention that multi-phase strategies have also been employed in interior-point techniques; e.g., see [14, 59].

3.2 Algorithm Description

We present our algorithm in the context of the generic nonlinear constrained optimization problem

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad (\min_x) \quad f(x) \\ &\text{subject to} \quad (\text{s.t.}) \quad c_{\mathcal{E}}(x) = 0, \quad c_{\mathcal{I}}(x) \leq 0, \end{aligned} \tag{3.2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $c_{\mathcal{E}} : \mathbb{R}^n \to \mathbb{R}^{m_{\mathcal{E}}}$, and $c_{\mathcal{I}} : \mathbb{R}^n \to \mathbb{R}^{m_{\mathcal{I}}}$ are twice-continuously differentiable. If the constraints of (3.2.1) are infeasible, then the algorithm is designed to return an infeasibility certificate in the form of a minimizer of the ℓ_1 infeasibility measure of the constraints; i.e., in such cases it is designed to solve

$$\min_x \ v(x), \quad \text{where} \quad v(x) := \|c_{\mathcal{E}}(x)\|_1 + \|[c_{\mathcal{I}}(x)]^+\|_1. \tag{3.2.2}$$

Here, for a vector c, we define $[c]^+ := \max\{c, 0\}$ and, for future reference, define $[c]^- := \max\{-c, 0\}$ (both component-wise). The priority is to locate a stationary point for (3.2.1), but in all cases the algorithm is at least guaranteed to find a stationary point for (3.2.2), i.e., a stationary point for v. We say a point x is stationary for v if $0 \in \partial v(x)$, where $\partial v(x)$ is the Clarke subdifferential of v at x [6, 24] (see [7] for a complete review of first-order theory for potentially infeasible problems).

Each iteration of our algorithm consists of solving at most two QO subproblems, updating a penalty parameter, and performing a line search on an exact penalty function. In this regard, the method is broadly similar to that proposed in [7]; however, the algorithm contains numerous refinements included to ensure rapid local convergence in both feasible and infeasible cases. In this section, we present the details of each step of the algorithm.

Of particular importance is the integration of our penalty parameter updates around the QO solves as this parameter is critical for driving fast local convergence for infeasible instances. A complete description of our algorithm is presented at the end of this section.

We begin by describing the conditions under which our algorithm terminates finitely. In short, the algorithm continues iterating unless a stationary point for problem (3.2.1) has been found. We define such stationary points according to first-order optimality conditions for problems (3.2.1) and (3.2.2), all of which can be presented by utilizing the Fritz John (FJ) function for (3.2.1), namely

$$\mathcal{F}(x,\rho,\lambda) := \rho f(x) + {\lambda_{\mathcal{E}}}^T c_{\mathcal{E}}(x) + {\lambda_{\mathcal{I}}}^T c_{\mathcal{I}}(x).$$

Here, $\rho \in \mathbb{R}$ is an objective multiplier and λ, with $\lambda_{\mathcal{E}} \in \mathbb{R}^{m_{\mathcal{E}}}$ and $\lambda_{\mathcal{I}} \in \mathbb{R}^{m_{\mathcal{I}}}$, are constraint multipliers. For future reference, we note that ρ also plays the role of the penalty parameter in the ℓ_1 exact penalty function

$$\phi(x,\rho) := \rho f(x) + v(x). \tag{3.2.3}$$

Our algorithm updates ρ and seeks stationary points for (3.2.1) through decreases in ϕ.

One possibility for finite termination is that the algorithm locates a first-order optimal point for (3.2.1). First-order optimality conditions for problem (3.2.1) are

$$\begin{aligned} \nabla_x \mathcal{F}(x,\rho,\lambda) = \rho \nabla f(x) + \nabla c_{\mathcal{E}}(x)\lambda_{\mathcal{E}} + \nabla c_{\mathcal{I}}(x)\lambda_{\mathcal{I}} = 0, \\ c_{\mathcal{E}}(x) = 0, \quad c_{\mathcal{I}}(x) \leq 0, \\ \lambda_{\mathcal{I}} \geq 0, \quad \lambda_{\mathcal{I}} \cdot c_{\mathcal{I}}(x) = 0. \end{aligned} \tag{3.2.4}$$

Here, $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is the gradient of f, $[\nabla c_{\mathcal{E}}]^T : \mathbb{R}^n \to \mathbb{R}^{m_{\mathcal{E}} \times n}$ is the Jacobian of $c_{\mathcal{E}}$ (and similarly for $[\nabla c_{\mathcal{I}}]^T$), and for vectors a and b we denote their component-wise (i.e., Hadamard or Schur) product by $a \cdot b$, a vector with entries $(a \cdot b)_i = a_i b_i$. If (x^*, ρ^*, λ^*) with $(\rho^*, \lambda^*) \neq 0$ satisfies (3.2.4), then we call (x^*, ρ^*, λ^*) stationary for (3.2.1); in particular, it is a FJ point [53]. Of particular interest are those FJ points with $\rho^* > 0$ as these correspond to Karush-Kuhn-Tucker (KKT) points for (3.2.1) [55, 56].

The other possibility for finite termination is that the algorithm locates a stationary point for (3.2.2) that is infeasible for problem (3.2.1). Hereinafter, defining e as a vector of ones (whose size is determined by the context), first-order optimality conditions for problem (3.2.2) are

$$\begin{aligned} \nabla_x \mathcal{F}(x, 0, \lambda) = \nabla c_{\mathcal{E}}(x)\lambda_{\mathcal{E}} + \nabla c_{\mathcal{I}}(x)\lambda_{\mathcal{I}} = 0, \\ -e \leq \lambda_{\mathcal{E}} \leq e, \quad 0 \leq \lambda_{\mathcal{I}} \leq e, \\ (e + \lambda_{\mathcal{E}}) \cdot [c_{\mathcal{E}}(x)]^- = 0, \quad (e - \lambda_{\mathcal{E}}) \cdot [c_{\mathcal{E}}(x)]^+ = 0, \\ \lambda_{\mathcal{I}} \cdot [c_{\mathcal{I}}(x)]^- = 0, \quad (e - \lambda_{\mathcal{I}}) \cdot [c_{\mathcal{I}}(x)]^+ = 0. \end{aligned} \tag{3.2.5}$$

If (x^*, λ^*) satisfies (3.2.5) and $v(x^*) > 0$, then we call (x^*, λ^*) stationary for (3.2.1); in particular, it is an infeasible stationary point. Despite the fact that such a point is infeasible for (3.2.1), it is deemed stationary as first-order information indicates that no further improvement in minimizing constraint violation locally is possible.

We now describe our technique for computing a search direction and multiplier estimates, which involves the solution of the QO subproblems (3.2.7) and (3.2.9) below. Once the details of these subproblems have been specified, we will describe an updating strategy for the penalty parameter that is integrated around these QO solves.

At the beginning of iteration k, the algorithm assumes an iterate of the form

$$(x^k, \rho^k, \overline{\lambda}^k, \widehat{\lambda}^k) \text{ with } \rho^k > 0, \; -e \leq \overline{\lambda}^k_{\mathcal{E}} \leq e, \; 0 \leq \overline{\lambda}^k_{\mathcal{I}} \leq e, \text{ and } \widehat{\lambda}^k_{\mathcal{I}} \geq 0. \tag{3.2.6}$$

As all stationary points for (3.2.1) are necessarily stationary for the constraint violation measure v, we initiate computation in iteration k by seeking to measure the possible improvement in minimizing the following linearized model of v at x^k:

$$l(d; x^k) := \|c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T d\|_1 + \|[c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T d]^+\|_1.$$

Specifically, defining $H(x, \rho, \lambda)$ as an approximation for the Hessian of $\mathcal{F}$ at (x, ρ, λ), we solve the following QO subproblem whose solution we denote as $(\overline{d}^k, \overline{r}^k, \overline{s}^k, \overline{t}^k)$:

$$\min_{(d,r,s,t)} \ e^T(r+s) + e^T t + \tfrac{1}{2} d^T H(x^k, 0, \overline{\lambda}^k) d$$
$$\text{s.t.} \begin{cases} c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T d = & r - s \\ c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T d \le & t \\ (r, s, t) \ge & 0. \end{cases} \tag{3.2.7}$$

As shown in Lemma 3.3.2 in §3.3.1, this subproblem is always feasible and, if $H(x^k, 0, \overline{\lambda}^k)$ is positive definite, then the solution component $\overline{d}^k$ is unique. In addition, $\overline{d}^k$ yields a nonnegative reduction in $l(\cdot; x^k)$, i.e.,

$$\Delta l(\overline{d}^k; x^k) := l(0; x^k) - l(\overline{d}^k; x^k) \ge 0, \tag{3.2.8}$$

where equality holds if and only if x^k is stationary for v.

Upon solving subproblem (3.2.7) and setting

$$\overline{\lambda}^{k+1} \ \text{ with } \ -e \le \overline{\lambda}_{\mathcal{E}}^{k+1} \le e \ \text{ and } \ 0 \le \overline{\lambda}_{\mathcal{I}}^{k+1} \le e$$

as the optimal multipliers for the linearized equality and inequality constraints in (3.2.7), we check for termination at an infeasible stationary point. Specifically, we consider the constraint violation measure v and the following residual for (3.2.5):

$$\begin{aligned} \mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) := \max\{ & \|\nabla_x \mathcal{F}(x^k, 0, \overline{\lambda}^{k+1})\|_\infty, \\ & \|(e - \lambda_{\mathcal{E}}^{k+1}) \cdot [c_{\mathcal{E}}(x^k)]^+\|_\infty, \|(e + \lambda_{\mathcal{E}}^{k+1}) \cdot [c_{\mathcal{E}}(x^k)]^-\|_\infty, \\ & \|(e - \lambda_{\mathcal{I}}^{k+1}) \cdot [c_{\mathcal{I}}(x^k)]^+\|_\infty, \|\lambda_{\mathcal{I}}^{k+1} \cdot [c_{\mathcal{I}}(x^k)]^-\|_\infty\}. \end{aligned}$$

If $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0$ and $v(x^k) > 0$, then $(x^k, \overline{\lambda}^{k+1})$ is an infeasible stationary point. Otherwise, as shown in Lemma 3.3.2 in §3.3.1, it follows that either $v(x^k) = 0$ or $\overline{d}^k$ is a direction of strict descent for v from x^k.

Having measured, in a particular sense, the possible improvement in minimizing constraint violation by solving the QO subproblem (3.2.7), the algorithm solves a second QO subproblem that seeks optimality. Denoting $\mathcal{E}^k$ and $\mathcal{I}^k$ as the sets of constraints that are *linearly satisfied* at the solution of (3.2.7) (i.e., that have $\bar{r}_i^k = \bar{s}_i^k = 0$ for $i \in \mathcal{E}$ or $\bar{t}_i^k = 0$ for $i \in \mathcal{I}$, respectively), we require that the computed direction maintains this set of linearly satisfied constraints. The other *linearly violated* constraints in $\mathcal{E}_c^k \cup \mathcal{I}_c^k$ (where $\mathcal{E}_c^k := \mathcal{E} \setminus \mathcal{E}^k$ and $\mathcal{I}_c^k := \mathcal{I} \setminus \mathcal{I}^k$) remain relaxed with slack variables whose values are penalized in the subproblem objective. The value of the penalty parameter employed at this stage is the value for ρ^k immediately prior to this second phase subproblem, which for future notational convenience we denote as $\widehat{\rho}^k$. Overall, we solve the following regularized QO subproblem whose solution we denote as $(\widehat{d}^k, \widehat{r}^k_{\mathcal{E}_c^k}, \widehat{s}^k_{\mathcal{E}_c^k}, \widehat{t}^k_{\mathcal{I}_c^k})$:

$$\begin{aligned} \min_{(d, r_{\mathcal{E}_c^k}, s_{\mathcal{E}_c^k}, t_{\mathcal{I}_c^k})} \quad & \widehat{\rho}^k \nabla f(x^k)^T d + e^T (r_{\mathcal{E}_c^k} + s_{\mathcal{E}_c^k}) + e^T t_{\mathcal{I}_c^k} + \tfrac{1}{2} d^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k) d \\ \text{s.t.} \quad & \begin{cases} c_{\mathcal{E}^k}(x^k) + \nabla c_{\mathcal{E}^k}(x^k)^T d = 0, & c_{\mathcal{E}_c^k}(x^k) + \nabla c_{\mathcal{E}_c^k}(x^k)^T d = r_{\mathcal{E}_c^k} - s_{\mathcal{E}_c^k}, \\ c_{\mathcal{I}^k}(x^k) + \nabla c_{\mathcal{I}^k}(x^k)^T d \le 0, & c_{\mathcal{I}_c^k}(x^k) + \nabla c_{\mathcal{I}_c^k}(x^k)^T d \le t_{\mathcal{I}_c^k}, \\ & (r_{\mathcal{E}_c^k}, s_{\mathcal{E}_c^k}, t_{\mathcal{I}_c^k}) \ge 0. \end{cases} \end{aligned} \tag{3.2.9}$$

Upon solving (3.2.9) and setting

$$\widehat{\lambda}^{k+1} \ \text{ with } \ -e \le \widehat{\lambda}^{k+1}_{\mathcal{E}_c^k} \le e, \ 0 \le \widehat{\lambda}^{k+1}_{\mathcal{I}_c^k} \le e, \ \text{ and } \ \widehat{\lambda}^{k+1}_{\mathcal{I}^k} \ge 0$$

as the optimal multipliers for the linearized equality and inequality constraints in (3.2.9), it is again appropriate to check for finite termination of the algorithm, this time with respect to the optimality conditions for (3.2.1). Given $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ we consider the violation measure v and the following residual corresponding to (3.2.4):

$$\mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) := \max\{\|\nabla_x \mathcal{F}(x^k, \rho^k, \widehat{\lambda}^{k+1})\|_\infty, \|\widehat{\lambda}^{k+1}_{\mathcal{I}} \cdot c_{\mathcal{I}}(x^k)\|_\infty\}.$$

We prove in Lemma 3.3.4 in §3.3.1 that if the algorithm reaches this stage, then ρ^k is strictly positive. Thus, if $\mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) = 0$ and $v(x^k) = 0$, then $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ is a KKT point for (3.2.1).

If the algorithm has not terminated finitely due to this last check of optimality, then the search direction d^k is chosen as a convex combination of the directions obtained from subproblems (3.2.7) and (3.2.9). Given a constant $\beta \in (0, 1)$, our criterion for the selection of the weights in this combination is

$$\Delta l(d^k; x^k) \geq \beta \Delta l(\overline{d}^k; x^k). \tag{3.2.10}$$

For $w \in [0, 1]$, the reduction in $l(\cdot; x^k)$ obtained by

$$d(w) := w\overline{d}^k + (1 - w)\widehat{d}^k \tag{3.2.11}$$

is a piecewise linear function of w. If $\Delta l(\overline{d}^k; x^k) = 0$, then by the formulation of (3.2.9), we have $\Delta l(\widehat{d}^k; x^k) = 0$ and so (3.2.10) is satisfied by $w = 0$. Otherwise, if $\Delta l(\overline{d}^k; x^k) > 0$, then since $\Delta l(d(1); x^k) = \Delta l(\overline{d}^k; x^k) > \beta \Delta l(\overline{d}^k; x^k)$, there exists a threshold $\underline{w} \in [0, 1)$ such that (3.2.10) holds for all $w \geq \underline{w}$. We define w^k as the smallest value in $[0, 1)$ such that (3.2.10) holds and set the search direction as $d^k \leftarrow d(w^k)$.

We have presented our techniques for computing the primal search direction d^k as well as new multiplier estimates $\overline{\lambda}^{k+1}$ and $\widehat{\lambda}^{k+1}$. Within this discussion, we have accounted for finite termination of the algorithm and highlighted certain consequences of our step computation procedure (e.g., (3.2.8) and (3.2.10)) that will be critical in our convergence analysis. All that remains in the specification of our algorithm is our updating strategy for the penalty parameter and the conditions of our line search, which we now present. Note that with respect to ρ, an update is considered twice in a given iteration. The first time an update is considered is between the two QO subproblem solves, as it is at this point in the algorithm where the solution of (3.2.7) may trigger aggressive action toward infeasibility detection. The second time an update is considered is after the solution of

(3.2.9). The update considered at that time is representative of typical contemporary updating strategies, used to ensure a well-defined line search and global convergence of the algorithm.

Prior to solving the second subproblem (3.2.9) (and before fixing $\widehat{\rho}^k$), we potentially modify ρ^k and $\widehat{\lambda}^k$ (computed in iteration $k-1$) to reduce the weight of the objective f and promote fast infeasibility detection. (Note that ρ^k and $\widehat{\lambda}^k$ will both influence the objective of (3.2.9).) If the current iterate is infeasible and the reduction in linearized feasibility obtained by $\overline{d}^k$ is small compared to the level of nonlinear infeasibility, then there is evidence that the algorithm is converging to an infeasible stationary point. In such cases, we consider modifying ρ^k before solving subproblem (3.2.9) so that the rest of the iteration places a higher emphasis on reducing constraint violation. A corresponding modification to $\widehat{\lambda}^k$ is also necessary to guarantee fast infeasibility detection (see Theorem 3.3.11). Defining constants $\theta \in (0,1)$, $\kappa_\rho > 0$, and $\kappa_\lambda > 0$, if

$$v(x^k) > 0 \quad \text{and} \quad \Delta l(\overline{d}^k; x^k) \le \theta v(x^k), \tag{3.2.12}$$

then we set ρ^k by

$$\rho^k \leftarrow \min\{\rho^k, \kappa_\rho \mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1})_2\} \tag{3.2.13}$$

and modify $\widehat{\lambda}^k$ so that

$$\|\widehat{\lambda}^k - \overline{\lambda}^k\| \le \kappa_\lambda \mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1})_2. \tag{3.2.14}$$

Otherwise, we maintain the current ρ^k and $\widehat{\lambda}^k$. For satisfying (3.2.14), a simple approach is to set $\widehat{\lambda}^k \leftarrow \alpha_\lambda \widehat{\lambda}^k + (1-\alpha_\lambda)\overline{\lambda}^k$ where α_λ is the largest value in $[0,1]$ such that (3.2.14) is satisfied. (This is the approach taken in our implementation described in §5.6.)

Upon solving (3.2.9) and assuming the algorithm does not immediately terminate, we turn to a second update for ρ and our line search. For these purposes, we employ the ℓ_1 exact penalty function ϕ (recall (3.2.3)). At x^k, a linear model of $\phi(\cdot, \rho)$ is

$$m(d; x^k, \rho) := \rho(f(x^k) + \nabla f(x^k)^T d) + l(d; x^k)$$

and the corresponding reduction in this model yielded by the search direction d^k is

$$\Delta m(d^k; x^k, \rho) := m(0; x^k, \rho) - m(d^k; x^k, \rho) = -\rho \nabla f(x^k)^T d^k + \Delta l(d^k; x^k). \tag{3.2.15}$$

Prior to the line search, the new penalty parameter ρ^{k+1} is set so that its reciprocal is larger than the largest multiplier (derived from (3.2.9)) and that the reduction $\Delta m(d^k; x^k, \rho^{k+1})$ is at least proportional to $\Delta l(d^k; x^k)$. That is, we set ρ^{k+1} so that

$$\rho^{k+1} \|\widehat{\lambda}^{k+1}\|_\infty \leq 1 \tag{3.2.16}$$

and, for a given constant $\epsilon \in (0, 1)$, we have

$$\Delta m(d^k; x^k, \rho^{k+1}) \geq \epsilon \Delta l(d^k; x^k). \tag{3.2.17}$$

Given constants $\delta \in (0, 1)$ and $\omega \in (0, 1)$, (3.2.16) and (3.2.17) can be achieved by setting

$$\rho^k \leftarrow \min\left\{\delta \rho^k, \frac{(1-\epsilon)}{\|\widehat{\lambda}^{k+1}\|_\infty}\right\} \quad \text{if} \quad \rho^k \|\widehat{\lambda}^{k+1}\|_\infty > 1 \tag{3.2.18}$$

followed by

$$\rho^k \leftarrow \begin{cases} \delta \rho^k & \text{if } \Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k) \text{ and } w^k \geq \omega; \\ \min\left\{\delta \rho^k, \zeta^k\right\} & \text{if } \Delta m(d^k; x^k, \rho^k) < \epsilon \Delta l(d^k; x^k), \end{cases} \tag{3.2.19}$$

where

$$\zeta^k := \frac{(1-\epsilon)\Delta l(d^k; x^k)}{\nabla f(x^k)^T d^k + \frac{1}{2}(d^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k) d^k},$$

and then setting $\rho^{k+1} \leftarrow \rho^k$. Once ρ^{k+1} has been set in this manner, we perform a backtracking line search along d^k to determine α^k such that, for $\eta \in (0, 1)$, we have

$$\phi(x^k + \alpha^k d^k, \rho^{k+1}) - \phi(x^k, \rho^{k+1}) \leq -\eta \alpha^k \Delta m(d^k; x^k, \rho^{k+1}). \tag{3.2.20}$$

Our proposed algorithm, hereinafter nicknamed `SQuID` , is presented as Algorithm 1. We claim that the algorithmic framework of `SQuID` is globally convergent for choices of subproblems other than (3.2.7). For instance, a linear subproblem with a trust region would be appropriate for determining the best local improvement in linearized feasibility; e.g., see [7, 8]. Under certain common assumptions, this choice should also allow for rapid local convergence for *feasible* problem instances. We present `SQuID` as solving two QO subproblems per iteration, however, as this choice also allows for rapid local convergence for *infeasible* instances, the main focus of this chapter. In particular, in the neighborhood of an infeasible stationary point satisfying the assumptions of §3.3.3, it can be seen that as $\rho^k \to 0$ and $\widehat{\lambda}^k \to \overline{\lambda}^k$, subproblem (3.2.9) produces SQO-like steps for the minimization of constraint violation, thus causing rapid convergence toward stationary points for v. This being said, efficient implementations of `SQuID` may avoid two QO solves per iteration. For example, at (nearly) feasible points, one may consider skipping subproblem (3.2.7) entirely, as we do in our implementation described in §5.6. For the purposes of this chapter, however, we analyze the behavior of `SQuID` as it has been presented.

Algorithm 1 *S*equential *Qu*adratic Optimizer with Rapid *I*nfeasibility *D*etection

1: Choose $\beta \in (0,1)$, $\theta \in (0,1)$, $\kappa_\rho > 0$, $\kappa_\lambda > 0$, $\epsilon \in (0,1)$, $\omega \in (0,1)$, $\delta \in (0,1)$, $\eta \in (0,1)$, and $\gamma \in (0,1)$. Set $k \leftarrow 0$ and choose $(x^k, \rho^k, \overline{\lambda}^k, \widehat{\lambda}^k)$ satisfying (3.2.6).

2: Compute $(\overline{d}^k, \overline{r}^k, \overline{s}^k, \overline{t}^k, \overline{\lambda}^{k+1})$ as the optimal primal-dual solution for (3.2.7).

3: If $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0$ and $v(x^k) > 0$, then terminate; $(x^k, \overline{\lambda}^{k+1})$ is an infeasible stationary point for problem (3.2.1).

4: If (3.2.12) holds, then set ρ^k by (3.2.13) and $\widehat{\lambda}^k$ so that (3.2.14) holds. Set $\widehat{\rho}^k \leftarrow \rho^k$.

5: Compute $(\widehat{d}^k, \widehat{r}^k_{\mathcal{E}_c^k}, \widehat{s}^k_{\mathcal{E}_c^k}, \widehat{t}^k_{\mathcal{I}_c^k}, \widehat{\lambda}^{k+1})$ as the optimal primal-dual solution for (3.2.9).

6: If $\mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) = 0$ and $v(x^k) = 0$, then terminate; $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ is a KKT point for problem (3.2.1).

7: Set d^k by (3.2.11) where w^k is the smallest value in $[0,1)$ such that (3.2.10) holds.

8: Update ρ^k by (3.2.18), then by (3.2.19), and finally set $\rho^{k+1} \leftarrow \rho^k$.

9: Let α^k be the largest value in $\{\gamma_0, \gamma_1, \gamma_2, \dots\}$ such that (3.2.20) holds.

10: Set $x^{k+1} \leftarrow x^k + \alpha^k d^k$ and $k \leftarrow k+1$ and go to step 2.

3.3 Convergence Analysis

The convergence properties of `SQuID` are the subject of this section. We prove the well-posedness of the algorithm along with global and local convergence results for feasible and infeasible problem instances. A few of the earlier results in this section are well-known in (nonsmooth) composite function theory, so for the sake of brevity we only provide citations for proofs.

3.3.1 Well-Posedness

We prove that `SQuID` is well-posed in that each iteration is well-defined and, if the overall algorithm does not terminate finitely, then an infinite sequence of iterates will be produced. This can be guaranteed under the following assumption. (Note that for simplicity here and in §3.3.2, we assume that subproblems (3.2.7) and (3.2.9) are convex. See §3.3.3 for a discussion of how this assumption can be relaxed without sacrificing local superlinear convergence guarantees.)

Assumption 3.3.1. *The following hold true for the iterates generated by* `SQuID` *:*

(a) The problem functions f, $c_{\mathcal{E}}$, and $c_{\mathcal{I}}$ are continuously differentiable in an open convex set containing $\{x^k\}$ and $\{x^k + d^k\}$.

(b) For all k, $H(x^k, 0, \overline{\lambda}^k)$ and $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$ are positive definite.

Our first lemma reveals that $-\Delta l(d; x^k)$ and $-\Delta m(d; x^k, \rho)$ respectively play the roles of surrogates for the directional derivatives of v and $\phi(\cdot.\rho)$ from x^k along the direction d. For a proof, see [6, Lemma 2.3]. We use the lemma to show that as long as a search direction d^k yields a strictly positive reduction in $l(\cdot, x^k)$ ($m(\cdot; x^k, \rho)$), then it is a direction of strict decrease for v ($\phi(\cdot, \rho)$).

Lemma 3.3.1. *The reductions in $l(\cdot; x^k)$ and $m(\cdot; x^k, \rho)$ produced by d satisfy*

$$Dv(d; x^k) \leq -\Delta l(d; x^k) \quad \text{and} \quad D\phi(d; x^k, \rho) \leq -\Delta m(d; x^k, \rho), \tag{3.3.1}$$

where $Dv(d;x^k)$ and $D\phi(d;x^k,\rho)$ represent the directional derivatives of v and $\phi(\cdot,\rho)$ at x^k corresponding to a step d, respectively.

The next lemma enumerates relevant properties of subproblem (3.2.7) related to the well-posedness of `SQuID` . It states that as long as x^k is not stationary for v, the solution component $\overline{d}^k$ will be a descent direction for v from x^k. These properties are well-known; e.g., see [6, Theorem 3.6].

Lemma 3.3.2. *Suppose Assumption 3.3.1 holds. Then, during iteration k of* `SQuID` *:*

(a) Subproblem (3.2.7) *is feasible and the solution component $\overline{d}^k$ is unique.*

(b) $\Delta l(\overline{d}^k;x^k) \geq 0$ where equality holds if and only if $\overline{d}^k = 0$.

(c) $\overline{d}^k = 0$ if and only if x^k is stationary for v.

(d) $\overline{d}^k = 0$ if and only if $(x^k, \overline{\lambda}^{k+1})$ satisfies (3.2.5).

Properties of subproblem (3.2.9) related to the well-posedness of `SQuID` are enumerated in the next lemma.

Lemma 3.3.3. *Suppose Assumption 3.3.1 holds. Then, during iteration k of* `SQuID` *:*

(a) Subproblem (3.2.9) *is feasible and the solution component $\widehat{d}^k$ is unique.*

(b) With $\rho^k > 0$ and $v(x^k) = 0$, step 5 yields $\widehat{d}^k = 0$ if and only if $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ is a KKT point for (3.2.1).

Proof. By straightforward verification of the constraint function values, it follows that $\overline{d}^k$ is feasible for subproblem (3.2.9). Moreover, as $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$ is positive definite under Assumption 3.3.1, the objective of (3.2.9) is strictly convex and bounded below over the feasible set of the subproblem. Together, these statements imply that subproblem (3.2.9) is feasible and that the solution component $\widehat{d}^k$ is unique. This proves part (a).

For part (b), the solution $(\widehat{d}^k, \widehat{r}^k, \widehat{s}^k, \widehat{t}^k, \widehat{\lambda}^{k+1})$ satisfies the KKT conditions

$$\rho^k \nabla f(x^k) + H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\widehat{d}^k + \nabla c_{\mathcal{E}}(x^k)\widehat{\lambda}_{\mathcal{E}}^{k+1} + \nabla c_{\mathcal{I}}(x^k)\widehat{\lambda}_{\mathcal{I}}^{k+1} = 0, \tag{3.3.2a}$$

$$c_{\mathcal{E}^k}(x^k) + \nabla c_{\mathcal{E}^k}(x^k)^T \widehat{d}^k = 0, \quad c_{\mathcal{E}_c^k}(x^k) + \nabla c_{\mathcal{E}_c^k}(x^k)^T \widehat{d}^k = \widehat{r}_{\mathcal{E}_c^k}^k - \widehat{s}_{\mathcal{E}_c^k}^k \tag{3.3.2b}$$

$$c_{\mathcal{I}^k}(x^k) + \nabla c_{\mathcal{I}^k}(x^k)^T \widehat{d}^k \le 0, \quad c_{\mathcal{I}_c^k}(x^k) + \nabla c_{\mathcal{I}_c^k}(x^k)^T \widehat{d}^k \le \widehat{t}_{\mathcal{I}_c^k}^k \tag{3.3.2c}$$

$$\widehat{\lambda}_{\mathcal{I}^k}^{k+1} \cdot (c_{\mathcal{I}^k}(x^k) + \nabla c_{\mathcal{I}^k}(x^k)^T \widehat{d}^k) = 0, \tag{3.3.2d}$$

$$\widehat{\lambda}_{\mathcal{I}_c^k}^{k+1} \cdot (c_{\mathcal{I}_c^k}(x^k) + \nabla c_{\mathcal{I}_c^k}(x^k)^T \widehat{d}^k - t_{\mathcal{I}_c^k}^k) = 0, \tag{3.3.2e}$$

$$(e - \widehat{\lambda}_{\mathcal{E}_c^k}^{k+1}) \cdot \widehat{r}_{\mathcal{E}_c^k}^k = 0, \quad (e + \widehat{\lambda}_{\mathcal{E}_c^k}^{k+1}) \cdot \widehat{s}_{\mathcal{E}_c^k}^k = 0, \quad (e - \widehat{\lambda}_{\mathcal{I}_c^k}^{k+1}) \cdot \widehat{t}_{\mathcal{I}_c^k}^k = 0, \tag{3.3.2f}$$

$$-e \le \widehat{\lambda}_{\mathcal{E}_c^k}^{k+1} \le e, \; 0 \le \widehat{\lambda}_{\mathcal{I}_c^k}^{k+1} \le e, \text{ and } \widehat{\lambda}_{\mathcal{I}^k}^{k+1} \ge 0 \tag{3.3.2g}$$

from which it is easily shown that

$$\widehat{r}_{\mathcal{E}_c^k}^k = [c_{\mathcal{E}_c^k}(x^k) + \nabla c_{\mathcal{E}_c^k}(x^k)^T \widehat{d}^k]^+, \quad \widehat{s}_{\mathcal{E}_c^k}^k = [c_{\mathcal{E}_c^k}(x^k) + \nabla c_{\mathcal{E}_c^k}(x^k)^T \widehat{d}^k]^-,$$
$$\text{and } \widehat{t}_{\mathcal{I}_c^k}^k = [c_{\mathcal{I}_c^k}(x^k) + \nabla c_{\mathcal{I}_c^k}(x^k)^T \widehat{d}^k]^+.$$

Since we assume $v(x^k) = 0$, it follows that $(\overline{d}^k, \overline{r}^k, \overline{s}^k, \overline{t}^k) = 0$ is optimal for (3.2.7), which means $\mathcal{E}^k = \mathcal{E}$ and $\mathcal{I}^k = \mathcal{I}$. The optimality conditions (3.3.2) thus reduce to

$$\rho^k \nabla f(x^k) + H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\widehat{d}^k + \nabla c_{\mathcal{E}}(x^k)\widehat{\lambda}_{\mathcal{E}}^{k+1} + \nabla c_{\mathcal{I}}(x^k)\widehat{\lambda}_{\mathcal{I}}^{k+1} = 0, \tag{3.3.3a}$$

$$c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T \widehat{d}^k = 0, \quad c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T \widehat{d}^k \le 0, \tag{3.3.3b}$$

$$\widehat{\lambda}_{\mathcal{I}}^{k+1} \ge 0, \quad \widehat{\lambda}_{\mathcal{I}}^{k+1} \cdot (c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T \widehat{d}^k) = 0. \tag{3.3.3c}$$

Since we assume $\rho^k > 0$, by comparing the elements of $\mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1})$ with those of (3.3.3), it follows that $\widehat{d}^k = 0$ if and only if $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ is a KKT point for (3.2.1). $\square$

The next lemma shows that the updates for the penalty parameter in steps 4 and 8 are well-defined and that the latter update guarantees that $\Delta m(d^k; x^k, \rho^{k+1})$ is nonnegative. This can then be used to show, as we do in the lemma, that the line search in step 9 will terminate finitely with a positive step-size $\alpha^k > 0$.

Lemma 3.3.4. *Suppose Assumption 3.3.1 holds. Then, during iteration k of* `SQuID` *:*

(a) If at the beginning of iteration k we have $\rho^k > 0$, then, after step 4, $\rho^k > 0$.

(b) If at the beginning of step 8 we have $\rho^k > 0$, then, after step 8, $\rho^{k+1} > 0$ and

$$\Delta m(d^k; x^k, \rho^{k+1}) \geq \epsilon \Delta l(d^k; x^k) \geq \beta \epsilon \Delta l(\overline{d}^k; x^k) \geq 0. \tag{3.3.4}$$

(c) The line search in step 9 terminates with $\alpha^k > 0$.

Proof. If at step 4 we have $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0$, then we must have $v(x^k) = 0$ or else `SQuID` would have terminated in step 3. Thus, since (3.2.12) does not hold, step 4 will maintain the current $\rho^k > 0$. On the other hand, if at step 4 we have $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) > 0$, then either ρ^k will be maintained at its current positive value or (3.2.13) will set $\rho^k > 0$. This proves part (a).

For part (b), first consider (3.2.18). If $\|\widehat{\lambda}^{k+1}\|_\infty = 0$, then $\rho^k \|\widehat{\lambda}^{k+1}\|_\infty < 1$, meaning that (3.2.18) will not trigger a reduction in ρ^k. On the other hand, if $\|\widehat{\lambda}^{k+1}\|_\infty > 0$, then (3.2.18) will only ever yield $\rho^k > 0$. Thus, after applying (3.2.18), we have $\rho^k > 0$. Now consider (3.2.19). We have $\Delta l(d^k; x^k) \geq \beta \Delta l(\overline{d}^k; x^k) \geq 0$ due to (3.2.8) and (3.2.10), so all that remains is to show that $\rho^{k+1} > 0$ and $\Delta m(d^k; x^k, \rho^{k+1}) \geq \epsilon \Delta l(d^k; x^k)$. There are two cases to consider: $\Delta l(d^k; x^k) = 0$ and $\Delta l(d^k; x^k) > 0$. If $\Delta l(d^k; x^k) = 0$, then according to (3.2.10) and Lemma 3.3.2 we must have $\overline{d}^k = 0$. Moreover, if $v(x^k) \neq 0$, then Lemma 3.3.2 implies that the algorithm would have terminated in step 3, so since we are in step 8, we must have $v(x^k) = 0$, $\mathcal{E}^k = \mathcal{E}$, and $\mathcal{I}^k = \mathcal{I}$. It follows that in step 7 we obtain $d^k = \widehat{d}^k$ (i.e., $w^k = 0$) satisfying $\nabla f(x^k)^T d^k \leq 0$. Observing (3.2.15), we find that $\Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k)$ and $w^k < \omega$, so a reduction in ρ^k is not triggered by (3.2.19), the algorithm sets $\rho^{k+1} \leftarrow \rho^k$, and (3.3.4) is satisfied. Now consider when $\Delta l(d^k; x^k) > 0$. If $\Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k)$ and $w^k < \omega$, then there is nothing left to prove as the algorithm sets $\rho^{k+1} \leftarrow \rho^k$ and (3.3.4) holds. If $\Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k)$, but $w^k \geq \omega$, then

$$\Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k) \implies \rho^k \nabla f(x^k)^T d^k \leq (1 - \epsilon) \Delta l(d^k; x^k).$$

Then, since $\Delta l(d^k; x^k) > 0$, it follows that prior to the update (3.2.19) we have

$$\delta\rho^k \nabla f(x^k)^T d^k \leq (1-\epsilon)\Delta l(d^k; x^k) \implies \Delta m(d^k; x^k, \delta\rho^k) \geq \epsilon \Delta l(d^k; x^k).$$

As a result, after the update (3.2.19), we again have that $\rho^{k+1} > 0$ and (3.3.4) holds. Finally, if $\Delta m(d^k; x^k, \rho^k) < \epsilon \Delta l(d^k; x^k)$, then by (3.2.15) we must have $\nabla f(x^k)^T d^k > 0$. In such cases, after ρ^k is updated by (3.2.19) (to a positive value since $\zeta^k > 0$), we have

$$\begin{aligned}
\Delta m(d^k; x^k, \rho^k) &= -\rho^k \nabla f(x^k)^T d^k + \Delta l(d^k; x^k) \\
&\geq -\frac{(1-\epsilon)\Delta l(d^k; x^k)}{\nabla f(x^k)^T d^k + \frac{1}{2}(d^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k) d^k} \nabla f(x^k)^T d^k + \Delta l(d^k; x^k) \\
&\geq (\epsilon - 1)\Delta l(d^k; x^k) + \Delta l(d^k; x^k) \\
&= \epsilon \Delta l(d^k; x^k),
\end{aligned}$$

completing the proof of part (b) of the lemma.

Finally, for part (c), we first claim that $\Delta m(d^k; x^k, \rho^{k+1}) > 0$ in step 9. Indeed, by part (b), the model reduction satisfies $\Delta m(d^k; x^k, \rho^{k+1}) = 0$ only if $\Delta l(\overline{d}^k; x^k) = 0$. However, by Lemma 3.3.2 and the formulation of (3.2.9), this occurs if and only if x^k is stationary for v. If $v(x^k) > 0$, then `SQuID` would have terminated in step 3; thus, we may assume $v(x^k) = 0$. Moreover, if $d^k = 0$, then by Lemma 3.3.3, `SQuID` would have terminated in step 6; thus, we may assume $d^k \neq 0$. Since under these conditions the point $(d, r, s, t) = (0, 0, 0, 0)$ is feasible for (3.2.9) and yields an objective value of 0 for that subproblem, we must have $\nabla f(x^k)^T d^k < 0$, meaning that $\Delta m(d^k; x^k, \rho^{k+1}) = -\rho^{k+1} \nabla f(x^k)^T d^k > 0$. Overall, we have shown that if the algorithm enters step 9, then $\Delta m(d^k; x^k, \rho^{k+1}) > 0$. This fact and Lemma 3.3.1 reveal that d^k is a direction of strict descent for $\phi(\cdot, \rho^{k+1})$ from x^k, implying that the backtracking line search will terminate with a positive step-size $\alpha^k > 0$. □

Our main theorem in this subsection summarizes the well-posedness of `SQuID` .

Theorem 3.3.2. *Suppose Assumption 3.3.1 holds. Then, one of the following holds:*

(a) *SQuID terminates in step 3 with* $(x^k, \overline{\lambda}^{k+1})$ *satisfying*

$$\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0 \quad and \quad v(x^k) > 0;$$

(b) *SQuID terminates in step 6 with* $(x^k, \rho^k, \widehat{\lambda}^{k+1})$ *satisfying*

$$\rho^k > 0, \quad \mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) = 0, \quad and \quad v(x^k) = 0;$$

(c) *SQuID generates an infinite sequence* $\{(x^k, \rho^k, \overline{\lambda}^k, \widehat{\lambda}^k)\}$ *where, for all* k*,*

$$\rho^k > 0,\ -e \le \overline{\lambda}^k_{\mathcal{E}} \le e,\ 0 \le \overline{\lambda}^k_{\mathcal{I}} \le e,\ -e \le \widehat{\lambda}^k_{\mathcal{E}^k_c} \le e,\ 0 \le \widehat{\lambda}^k_{\mathcal{I}^k_c} \le e,\ and\ \widehat{\lambda}^k_{\mathcal{I}^k} \ge 0.$$

Proof. By Lemmas 3.3.2, 3.3.3, and 3.3.4, each iteration of SQuID terminates finitely. If SQuID itself does not terminate finitely in step 3 or 6, then steps 2 and 5 and the optimality conditions for subproblems (3.2.7) and (3.2.9) yield the bounds in statement (c). Moreover, by Lemma 3.3.4(a)–(b), it follows that an infinite number of SQuID iterates yields $\{\rho^k\} > 0$. □

3.3.2 Global Convergence

We now prove properties related to the global convergence of SQuID under the assumption that an infinite sequence of iterates is generated; i.e., we focus on the situation described in Theorem 3.3.2(c). These properties require a slight strengthening of our assumptions from §3.3.1. (As Assumption 3.3.3 is stronger than Assumption 3.3.1, it follows that all results from §3.3.1 still hold.)

Assumption 3.3.3. *The following hold true for the iterates generated by SQuID :*

(a) *The problem functions* f*,* $c_{\mathcal{E}}$*,* $c_{\mathcal{I}}$ *and their first derivatives are bounded and Lipschitz continuous in an open convex set containing* $\{x^k\}$ *and* $\{x^k + d^k\}$*.*

(b) There exist constants $\overline{\mu} \geq \underline{\mu} > 0$ such that, for all k and $d \in \mathbb{R}^n$,

$$\underline{\mu}\|d\|_2 \leq d^T H(x^k, 0, \overline{\lambda}^k) d \leq \overline{\mu}\|d\|_2 \quad \text{and} \quad \underline{\mu}\|d\|_2 \leq d^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k) d \leq \overline{\mu}\|d\|_2.$$

Of particular interest at the end of this section is the behavior of `SQuID` in the vicinity of points satisfying the Mangasarian-Fromovitz constraint qualification (MFCQ) for problem (3.2.1). We define this well-known qualification for convenience.

Definition 3.3.1. *A point x satisfies the MFCQ for problem* (3.2.1) *if $v(x) = 0$, $\nabla c_{\mathcal{E}}(x)$ has full column rank, and there exists $d \in \mathbb{R}^n$ such that*

$$c_{\mathcal{E}}(x) + \nabla c_{\mathcal{E}}(x)^T d = 0 \quad \text{and} \quad c_{\mathcal{I}}(x) + \nabla c_{\mathcal{I}}(x)^T d < 0.$$

In this and the following subsection, at x^k, let the sets of positive, zero, and negative-valued equality constraints be defined, respectively, as

$$\mathcal{P}^k := \{i \in \mathcal{E} : c_i(x^k) > 0\}, \ \mathcal{Z}^k := \{i \in \mathcal{E} : c_i(x^k) = 0\}, \ \text{and } \mathcal{N}^k := \{i \in \mathcal{E} : c_i(x^k) < 0\}.$$

Similarly, let the sets of violated, active, and strictly satisfied inequality constraints, respectively, be

$$\mathcal{V}^k := \{i \in \mathcal{I} : c_i(x^k) > 0\}, \ \mathcal{A}^k := \{i \in \mathcal{I} : c_i(x^k) = 0\}, \ \text{and } \mathcal{S}^k := \{i \in \mathcal{I} : c_i(x^k) < 0\}.$$

We similarly define the sets $\mathcal{P}^*$, $\mathcal{Z}^*$, $\mathcal{N}^*$, $\mathcal{V}^*$, $\mathcal{A}^*$, and $\mathcal{S}^*$ when referring to those index sets corresponding to a point of interest x^*.

The following lemma shows that the norms of the search directions are bounded. This result can also be seen to follow if one applies [6, Lemma 3.4].

Lemma 3.3.5. *Suppose Assumption 3.3.3 holds. Then, the sequences $\{\|\overline{d}^k\|\}$ and $\{\|\widehat{d}^k\|\}$ are bounded above, so the sequence $\{\|d^k\|\}$ is bounded above.*

Proof. Under Assumption 3.3.3, there exists $\tau > 0$ such that $v(x^k) \leq \tau$ for any k. In order to derive a contradiction to the statement in the lemma, suppose that $\{\|\overline{d}^k\|\}$ is not bounded. Then, there exists an iteration k yielding $\|\overline{d}^k\|_2 > 2\tau/\underline{\mu}$. The objective value of subproblem (3.2.7) corresponding to this $\overline{d}^k$ satisfies

$$l(\overline{d}^k; x^k) + \tfrac{1}{2}(\overline{d}^k)^T H(x^k, 0, \overline{\lambda}^k)\overline{d}^k \geq \tfrac{1}{2}\underline{\mu}\|\overline{d}^k\|_2 > \tau \geq v(x^k).$$

However, this is a contradiction as $v(x^k)$ is the objective value corresponding to $(d, r, s, t) = (0, [c_{\mathcal{E}}(x^k)]^+, [c_{\mathcal{E}}(x^k)]^-, [c_{\mathcal{I}}(x^k)]^+)$, which is also feasible for this subproblem. Thus, $\|\overline{d}^k\|_2 \leq 2\tau/\underline{\mu}$ for all k, so $\{\|\overline{d}^k\|\}$ is bounded.

Now suppose, in order to derive a different contradiction, that for some k the optimal solution for (3.2.9) yields $\underline{\mu}\|\widehat{d}^k\| > 8\rho_0\|\nabla f(x^k)\|$ and $\underline{\mu}\|\widehat{d}^k\|_2 > 2\overline{\mu}\|\overline{d}^k\|_2$. Then, under Assumption 3.3.3, we find

$$\begin{aligned}
&- \rho^k \nabla f(x^k)^T \widehat{d}^k + \rho^k \nabla f(x^k)^T \overline{d}^k + \tfrac{1}{2}(\overline{d}^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\overline{d}^k \\
&\leq \rho_0 \|\nabla f(x^k)\|\|\widehat{d}^k\| + \rho_0\|\nabla f(x^k)\|\|\overline{d}^k\| + \tfrac{1}{2}\overline{\mu}\|\overline{d}^k\|_2 \\
&< \tfrac{1}{8}\underline{\mu}\|\widehat{d}^k\|_2 + \tfrac{1}{8}\underline{\mu}\sqrt{\frac{\underline{\mu}}{2\overline{\mu}}}\|\widehat{d}^k\|_2 + \tfrac{1}{4}\underline{\mu}\|\widehat{d}^k\|_2 \\
&\leq \tfrac{1}{2}\underline{\mu}\|\widehat{d}^k\|_2 \\
&\leq \tfrac{1}{2}(\widehat{d}^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\widehat{d}^k.
\end{aligned}$$

Since $(\overline{d}^k, \overline{r}^k, \overline{s}^k, \overline{t}^k)$ is feasible for (3.2.9) and the above implies

$$\rho^k \nabla f(x^k)^T \widehat{d}^k + \tfrac{1}{2}(\widehat{d}^k)^T H(x^k, \rho^k, \widehat{\lambda}^k)\widehat{d}^k > \rho^k \nabla f(x^k)^T \overline{d}^k + \tfrac{1}{2}(\overline{d}^k)^T H(x^k, \rho^k, \widehat{\lambda}^k)\overline{d}^k,$$

it follows that $(\widehat{d}^k, \widehat{r}^k, \widehat{s}^k, \widehat{t}^k)$ cannot be the optimal solution for (3.2.9), a contradiction. Thus, for all k,

$$\|\widehat{d}^k\| \leq \max\left\{8\rho_0\|\nabla f(x^k)\|/\underline{\mu},\, \sqrt{2\overline{\mu}/\underline{\mu}}\|\overline{d}^k\|\right\}$$

and since $\{\|\overline{d}^k\|\}$ and $\{\nabla f(x^k)\}$ are bounded by the above paragraph and Assumption 3.3.3, respectively, it follows that $\{\|\widehat{d}^k\|\}$ is also bounded.

The boundedness of $\{\|d^k\|\}$ follows from the above results and the fact that d^k is chosen as a convex combination of $\overline{d}^k$ and $\widehat{d}^k$ for all k. $\square$

We also have the following lemma, providing a lower bound for α^k for each k.

Lemma 3.3.6. *Suppose Assumption 3.3.3 holds. Then, for all k, the stepsize satisfies $\alpha^k \geq c\Delta m(d^k; x^k, \rho^{k+1})$ for some constant $c > 0$ independent of k.*

Proof. Under Assumption 3.3.3, applying Taylor's Theorem and Lemma 3.3.1, we have that for all positive α that are sufficiently small, there exists $\tau > 0$ such that

$$\phi(x^k + \alpha d^k, \rho^{k+1}) - \phi(x^k, \rho^{k+1}) \leq -\alpha\Delta m(d^k; x^k, \rho^{k+1}) + \tau\alpha_2\|d^k\|_2.$$

Thus, for any $\alpha \in [0, (1-\eta)\Delta m(d^k; x^k, \rho^{k+1})/(\tau\|d^k\|_2)]$, we have

$$-\alpha\Delta m(d^k; x^k, \rho^{k+1}) + \tau\alpha_2\|d^k\|_2 \leq -\alpha\eta\Delta m(d^k; x^k, \rho^{k+1}),$$

meaning that the sufficient decrease condition (3.2.20) holds. During the line search, the stepsize is multiplied by γ until (3.2.20) holds, so we know by the above that the backtracking procedure terminates with

$$\alpha^k \geq \gamma(1-\eta)\Delta m(d^k; x^k, \rho^{k+1})/(\tau\|d^k\|_2).$$

The result follows from this inequality since, by Lemma 3.3.5, $\{\|d^k\|\}$ is bounded. $\square$

We now prove that, in the limit, the reductions in the models of the constraint violation measure and the penalty function vanish. For this purpose, it will be convenient to work with the shifted penalty function

$$\varphi(x, \rho) := \rho(f(x) - \underline{f}) + v(x) \geq 0, \tag{3.3.5}$$

where $\underline{f}$ is the infimum of f over the smallest convex set containing $\{x^k\}$. The existence of

$\underline{f}$ follows from Assumption 3.3.3. The function φ possesses a useful monotonicity property proved in the following lemma.

Lemma 3.3.7. *Suppose Assumption 3.3.3 holds. Then, for all k,*

$$\varphi(x^{k+1}, \rho^{k+2}) \leq \varphi(x^k, \rho^{k+1}) - \eta\alpha^k \Delta m(d^k; x^k, \rho^{k+1}),$$

so, by Lemmas 3.3.4 and 3.3.6, $\{\varphi(x^k, \rho^{k+1})\}$ is monotonically decreasing.

Proof. By the line search condition (3.2.20), we have

$$\varphi(x^{k+1}, \rho^{k+1}) \leq \varphi(x^k, \rho^{k+1}) - \eta\alpha^k \Delta m(d^k; x^k, \rho^{k+1}),$$

which implies

$$\varphi(x^{k+1}, \rho^{k+2}) \leq \varphi(x^k, \rho^{k+1}) - (\rho^{k+1} - \rho^{k+2})(f(x^{k+1}) - \underline{f}) - \eta\alpha^k \Delta m(d^k; x^k, \rho^{k+1}).$$

The result then follows from this inequality, the fact that $\{\rho^k\}$ is monotonically decreasing, and since $f(x^{k+1}) \geq \underline{f}$ for all k. □

We now show that the model reductions vanish in the limit.

Lemma 3.3.8. *Suppose Assumption 3.3.3 holds. Then, the following limits hold:*

$$0 = \lim_{k\to\infty} \Delta m(d^k; x^k, \rho^{k+1}) = \lim_{k\to\infty} \Delta l(d^k; x^k) = \lim_{k\to\infty} \Delta l(\overline{d}^k; x^k) = \lim_{k\to\infty} \Delta l(\widehat{d}^k; x^k).$$

Proof. In order to derive a contradiction, suppose that $\Delta m(d^k; x^k, \rho^{k+1})$ does not converge to 0. Then, by Lemma 3.3.4, there exists $\tau > 0$ and an infinite subsequence of iterates K such that $\Delta m(d^k; x^k, \rho^{k+1}) \geq \tau$ for all $k \in K$. By Lemmas 3.3.6 and 3.3.7, this would imply that $\varphi(x^k, \rho^{k+1}) \to -\infty$, which is impossible since $\{\varphi(x^k, \rho^{k+1})\}$ is bounded below by 0. Hence, $\Delta m(d^k; x^k, \rho^{k+1}) \to 0$. The other limits follow by Lemma 3.3.4(b), the fact that d^k is a convex combination of $\overline{d}^k$ and $\widehat{d}^k$ for all k, and the convexity of $\Delta l(\cdot; x^k)$ for all k. □

We now show that the primal solution components for the subproblems vanish in the limit, and thus the primal search directions vanish in the limit.

Lemma 3.3.9. *Suppose Assumption 3.3.3 holds. Then, the following limits hold:*

$$0 = \lim_{k\to\infty} \overline{d}^k = \lim_{k\to\infty} \widehat{d}^k = \lim_{k\to\infty} d^k.$$

Proof. First, we prove by contradiction that $\overline{d}^k \to 0$. Suppose there exists $\tau > 0$ and an infinite subsequence of iterates K such that $\|\overline{d}^k\| \geq \tau$ for all $k \in K$. By Lemma 3.3.8, there exists $k' \geq 0$ such that for all $k \geq k'$ we have $\Delta l(\overline{d}^k; x^k) \leq \underline{\mu}\tau_2/4$. (Recall that $\underline{\mu}$ is defined in Assumption 3.3.3.) Hence, we have that for some $k \in K$ such that $k \geq k'$, the optimal objective value of (3.2.7) satisfies

$$v(x^k) - \Delta l(\overline{d}^k; x^k) + \tfrac{1}{2}\overline{d}^{k^T} H(x^k, 0, \overline{\lambda}^k)\overline{d}^k \geq v(x^k) - \tfrac{1}{4}\underline{\mu}\tau_2 + \tfrac{1}{2}\underline{\mu}\tau_2 > v(x^k).$$

This is a contradiction as $v(x^k)$ is the objective value corresponding to

$$(d, r, s, t) = (0, [c_{\mathcal{E}}(x^k)]^+, [c_{\mathcal{E}}(x^k)]^-, [c_{\mathcal{I}}(x^k)]^+),$$

which is also feasible. Thus, $\overline{d}^k \to 0$.

Now we prove that $\widehat{d}^k \to 0$. To do this, we first prove that

$$\lim_{k\to\infty} \rho^k \nabla f(x^k)^T \widehat{d}^k = 0. \tag{3.3.6}$$

Indeed, (3.3.6) clearly holds if $\rho^k \to 0$ since $\{\nabla f(x^k)\}$ and $\{\widehat{d}^k\}$ are bounded by Assumption 3.3.3 and Lemma 3.3.5, respectively. Otherwise, if $\rho^k \nrightarrow 0$, then due to update (3.2.19) we must have $\Delta m(d^k; x^k, \rho^k) \geq \epsilon \Delta l(d^k; x^k)$ and $w^k < \omega$ for all sufficiently large k. Hence, by Lemma 3.3.8, (3.2.15), (3.2.11), the fact that $\{\rho^k\}$ is monotonically decreasing, the boundedness of $\{\nabla f(x^k)\}$ under Assumption 3.3.3, and $\overline{d}^k \to 0$, we have

$$\begin{aligned} 0 &= \lim_{k\to\infty} (\Delta l(d^k; x^k) - \Delta m(d^k; x^k, \rho^{k+1})) \\ &= \lim_{k\to\infty} \rho^{k+1} \nabla f(x^k)^T d^k \\ &= \lim_{k\to\infty} \rho^{k+1} \nabla f(x^k)^T (w^k \overline{d}^k + (1-w^k)\widehat{d}^k) \\ &= \lim_{k\to\infty} \rho^{k+1} (1-w^k) \nabla f(x^k)^T \widehat{d}^k. \end{aligned} \tag{3.3.7}$$

Since $(1 - w^k) > (1 - \omega) > 0$ for all sufficiently large k, and since $(\rho^{k+1} - \rho^k) \to 0$ follows from the facts that $\{\rho^k\}$ is monotonically decreasing and bounded below by zero, the limit (3.3.7) implies (3.3.6).

We may now use (3.3.6) to prove by contradiction that $\widehat{d}^k \to 0$. Suppose there exists $\tau > 0$ and an infinite subsequence of iterations K such that $\|\widehat{d}^k\| \geq \tau$ for all $k \in K$. By (3.3.6), there exists $k' \geq 0$ such that for all $k \geq k'$, we have $\rho^k \nabla f(x^k)^T \widehat{d}^k \geq -\underline{\mu}\tau_2/4$. Moreover, since $\overline{d}^k \to 0$, $\{\rho^k\}$ is monotonically decreasing, and $\{\nabla f(x^k)\}$ and $\{H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\}$ are bounded under Assumption 3.3.3, there exists $k'' \geq 0$ such that for all $k \geq k''$ we have

$$\rho^k \nabla f(x^k)^T \overline{d}^k < \tfrac{1}{16}\underline{\mu}\tau_2 \quad \text{and} \quad \tfrac{1}{2}(\overline{d}^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\overline{d}^k < \tfrac{1}{16}\underline{\mu}\tau_2. \tag{3.3.8}$$

Therefore, for $k \in K$ with $k \geq \max\{k', k''\}$, the above and Assumption 3.3.3(b) imply that the optimal objective value of (3.2.9) satisfies

$$\rho^k \nabla f(x^k)^T \widehat{d}^k + \tfrac{1}{2}\widehat{(}d^k)^T H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\widehat{d}^k \geq \tfrac{1}{4}\underline{\mu}\tau_2 > \rho^k \nabla f(x^k)^T \overline{d}^k + \frac{1}{2}(\overline{d}^k)^T H(x^k, \widehat{\rho}^k, \lambda^k)\overline{d}^k.$$

This contradicts the fact that $\widehat{d}^k$ is an optimal solution component of (3.2.9) since $(\overline{d}^k, \overline{r}^k, \overline{s}^k, \overline{t}^k)$ is feasible for (3.2.9) and the above implies that it yields a lower objective value than $(\widehat{d}^k, \widehat{r}^k, \widehat{s}^k, \widehat{t}^k)$. Hence, $\widehat{d}^k \to 0$.

The remainder of the result, namely that $d^k \to 0$, follows from the above and the fact that d^k is a convex combination of $\overline{d}^k$ and $\widehat{d}^k$ for all k. $\square$

We now present our first theorem of this subsection, which states that all limit points of a sequence generated by `SQuID` are first-order optimal for problem (3.2.2).

Theorem 3.3.4. *Suppose Assumption 3.3.3 holds. Then, the following limit holds:*

$$\lim_{k\to\infty} \mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0. \tag{3.3.9}$$

Therefore, all limit points of $\{(x^k, \overline{\lambda}^{k+1})\}$ *are first-order optimal for problem* (3.2.2).

Proof. Necessary and sufficient conditions for the optimality of $(\overline{d}^k, \overline{\lambda}^{k+1})$ with respect to (3.2.7) are

$$H(x^k, 0, \overline{\lambda}^k)\overline{d}^k + \nabla c_{\mathcal{E}}(x^k)\overline{\lambda}_{\mathcal{E}}^{k+1} + \nabla c_{\mathcal{I}}(x^k)\overline{\lambda}_{\mathcal{I}}^{k+1} = 0, \tag{3.3.10a}$$

$$-e \leq \overline{\lambda}_{\mathcal{E}}^{k+1} \leq e,\ 0 \leq \overline{\lambda}_{\mathcal{I}}^{k+1} \leq e, \tag{3.3.10b}$$

$$(e - \overline{\lambda}_{\mathcal{E}}^{k+1}) \cdot [c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T\overline{d}^k]^+ = 0, \tag{3.3.10c}$$

$$(e + \overline{\lambda}_{\mathcal{E}}^{k+1}) \cdot [c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T\overline{d}^k]^- = 0, \tag{3.3.10d}$$

$$(e - \overline{\lambda}_{\mathcal{I}}^{k+1}) \cdot [c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T\overline{d}^k]^+ = 0, \tag{3.3.10e}$$

$$\overline{\lambda}_{\mathcal{I}}^{k+1} \cdot [c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T\overline{d}^k]^- = 0, \tag{3.3.10f}$$

where we have eliminated

$$\overline{r}^k = [c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T\overline{d}^k]^+,\quad \overline{s}^k = [c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T\overline{d}^k]^-,$$
$$\text{and}\quad \overline{t}^k = [c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T\overline{d}^k]^+.$$

By Lemma 3.3.9, we have $\overline{d}^k \to 0$. Thus, as $\{H(x^k, 0, \overline{\lambda}^k)\}$, $\{\nabla c_{\mathcal{E}}(x^k)\}$, and $\{\nabla c_{\mathcal{I}}(x^k)\}$ are bounded under Assumption 3.3.3 and $\{\overline{\lambda}^{k+1}\}$ is bounded by (3.3.10b), it follows from (3.3.10) that $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) \to 0$. □

We now prove that if the penalty parameter remains bounded away from zero, then all feasible limit points of the iterate sequence correspond to KKT points.

Theorem 3.3.5. *Suppose Assumption 3.3.3 holds. Then, if* $\rho^k \to \rho^*$ *for some constant* $\rho^* > 0$ *and* $v(x^k) \to 0$*, the following limit holds:*

$$\lim_{k\to\infty} \mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) = 0.$$

Thus, every limit point (x^*, ρ^*, λ^*) *of* $\{(x^k, \rho^k, \widehat{\lambda}^{k+1})\}$ *with* $\rho^* > 0$ *and* $v(x^*) = 0$ *is a KKT point for problem* (3.2.1).

Proof. It follows from (3.3.2a) and Lemma 3.3.9 that under Assumption 3.3.3 we have

$$\nabla_x \mathcal{F}(x^k, \rho^k, \widehat{\lambda}^{k+1}) = -H(x^k, \rho^k, \widehat{\lambda}^k) d^k \to 0. \tag{3.3.11}$$

Thus, it only remains to show that $\widehat{\lambda}^k_{\mathcal{I}} \cdot c_{\mathcal{I}}(x^k) \to 0$ when $v(x^k) \to 0$. By Lemma 3.3.8 and the fact that

$$\Delta l(\widehat{d}^k; x^k) = v(x^k) - e^T(\widehat{r}^k_{\mathcal{E}^k_c} + \widehat{s}^k_{\mathcal{E}^k_c}) - e^T \widehat{t}^k_{\mathcal{I}^k_c} \quad \text{with} \quad (\widehat{r}^k_{\mathcal{E}^k_c}, \widehat{s}^k_{\mathcal{E}^k_c}, \widehat{t}^k_{\mathcal{I}^k_c}) \geq 0,$$

we have $\lim_{k\to\infty} \|\widehat{r}^k_{\mathcal{E}^k_c}\|_1 = \lim_{k\to\infty} \|\widehat{s}^k_{\mathcal{E}^k_c}\|_1 = \lim_{k\to\infty} \|\widehat{t}^k_{\mathcal{I}^k_c}\|_1 = 0$. If $\|\widehat{\lambda}^{k+1}\|_\infty$ is unbounded, then $\rho^k \to 0$ by (3.2.18), contradicting the conditions of the theorem. Hence, it follows from Lemma 3.3.9 that under Assumption 3.3.3 we have $(\widehat{d}^k)^T \nabla c_{\mathcal{I}}(x^k) \widehat{\lambda}^{k+1}_{\mathcal{I}} \to 0$. Consequently, from (3.3.2d) and (3.3.2e), we have

$$\begin{aligned} c_{\mathcal{I}^k_c}(x^k) \cdot \widehat{\lambda}^{k+1}_{\mathcal{I}^k_c} &= (\widehat{t}^k_{\mathcal{I}^k_c} - \nabla c_{\mathcal{I}^k_c}(x^k)^T \widehat{d}^k)^T \widehat{\lambda}^{k+1}_{\mathcal{I}^k_c} \to 0 \\ \text{and} \quad c_{\mathcal{I}^k}(x^k) \cdot \widehat{\lambda}^{k+1}_{\mathcal{I}^k} &= -\nabla c_{\mathcal{I}^k}(x^k)^T (\widehat{d}^k)^T \widehat{\lambda}^{k+1}_{\mathcal{I}^k} \to 0. \end{aligned} \tag{3.3.12}$$

The result follows from these limits and (3.3.11). □

We conclude this subsection with a theorem describing properties of limit points of `SQuID` whenever the penalty parameter vanishes.

Theorem 3.3.6. *Suppose Assumption 3.3.3 holds. Moreover, suppose* $\rho^k \to 0$ *and let* K_ρ *be the subsequence of iterations during which* ρ^k *is decreased by* (3.2.13), (3.2.18), *and/or*

(3.2.19). *Then, the following hold true:*

(a) Either all limit points of $\{x^k\}$ *are feasible for* (3.2.1) *or all are infeasible.*

(b) If all limit points of $\{x^k\}$ *are feasible, then all limit points of* $\{x^k\}_{k\in K_\rho}$ *correspond to FJ points for problem* (3.2.1) *where the MFCQ fails.*

Proof. For part (a), in order to derive a contradiction, suppose there exist infinite subsequences K^* and $K_\times$ such that $\{x^k\}_{k\in K^*} \to x^*$ with $v(x^*) = 0$ and $\{x^k\}_{k\in K_\times} \to x_\times$ with $v(x_\times) = \tau > 0$. Under Assumption 3.3.3 and since $\rho^k \to 0$, there exists $k^* \geq 0$ such that for all $k \in K^*$ with $k \geq k^*$ we have $\rho^{k+1}(f(x^k) - \underline{f}) < \tau/4$ and $v(x^k) < \tau/4$, meaning that $\varphi(x^k, \rho^{k+1}) < \tau/2$. (Recall that $\underline{f}$ has been defined as the infimum of f over the smallest convex set containing $\{x^k\}$.) On the other hand, we know that $\rho^{k+1}(f(x^k) - \underline{f}) \geq 0$ for all $k \geq 0$ and there exists $k_\times \geq 0$ such that for all $k \in K_\times$ with $k \geq k_\times$ we have $v(x^k) \geq \tau/2$, meaning that $\varphi(x^k, \rho^{k+1}) \geq \tau/2$. This is a contradiction since by Lemma 3.3.7 $\{\varphi(x^k, \rho^{k+1})\}$ is monotonically decreasing. Thus, the set of limit points of $\{x^k\}$ cannot include feasible and infeasible points at the same time.

For part (b), consider a subsequence $K^* \subseteq K_\rho$ such that $\{x^k\}_{k\in K^*} \to x^*$ for some limit point x^*. Let $K_1 \subseteq K^*$ be the subsequence of iterations during which ρ^k is decreased by (3.2.13) and let $K_2 \subseteq K^*$ be the subsequence of iterations during which it is decreased by (3.2.18) and/or (3.2.19). Since $K_1 \cup K_2 = K^*$ and K^* is infinite, it follows that K_1 or K_2 is infinite, or both. We complete the proof by considering two cases depending on the size of the index set K_2. In each case, our goal will be to show that a set of multipliers produced by `SQuID` have a nonzero limit point λ^* such that $(x^*, 0, \lambda^*)$ is a FJ point for problem (3.2.1). We then complete the proof by showing that the MFCQ fails at such limit points.

Case 1: Suppose K_2 is finite, meaning that for all sufficiently large k the algorithm does not decrease ρ^k in (3.2.18) nor in (3.2.19). Since $\{\overline{\lambda}^{k+1}\}_{k\in K_1}$ is bounded by (3.3.10b), it follows that this subsequence has a limit point. If all limit points of $\{\overline{\lambda}^{k+1}\}_{k\in K_1}$ are zero, then for all sufficiently large k we have $-e < \overline{\lambda}_{\mathcal{E}}^{k+1} < e$ and $0 \leq \overline{\lambda}_{\mathcal{I}}^{k+1} < e$. By (3.3.10c), (3.3.10d), and (3.3.10e), this implies

$$c_{\mathcal{E}}(x^k) + \nabla c_{\mathcal{E}}(x^k)^T \overline{d}^k = 0 \quad \text{and} \quad c_{\mathcal{I}}(x^k) + \nabla c_{\mathcal{I}}(x^k)^T \overline{d}^k \leq 0,$$

meaning that $\Delta l(\overline{d}^k; x^k) = v(x^k)$ for all such k. However, this implies that for all such k the algorithm does not decrease ρ^k by (3.2.13), implying that K_1 is also finite, a contradiction. Therefore, if K_2 is finite, then K_1 is infinite and there exists a nonzero limit point $\overline{\lambda}^*$ of $\{\overline{\lambda}^{k+1}\}_{k \in K_1}$.

Consider a subsequence $K_\lambda \subseteq K_1$ such that $\{(x^k, \overline{\lambda}^{k+1})\}_{k \in K_\lambda} \to (x^*, \overline{\lambda}^*)$. By Theorem 3.3.4, we have

$$\mathcal{R}_{opt}(x^*, 0, \overline{\lambda}^*) = \lim_{\substack{k \in K_\lambda \\ k \to \infty}} \mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) = 0,$$

meaning that $(x^*, 0, \overline{\lambda}^*)$ is a FJ point for problem (3.2.1).

Case 2: Suppose K_2 is infinite. We first prove that $\|\widehat{\lambda}^{k+1}\|_\infty > 1 - \epsilon$ for all sufficiently large $k \in K_2$. By contradiction, suppose there exists an infinite subsequence $K_\epsilon \subseteq K_2$ such that $\|\widehat{\lambda}^{k+1}\|_\infty \leq 1 - \epsilon$ for all $k \in K_\epsilon$. We will show that ρ^k will not be updated by (3.2.18) nor by (3.2.19), contradicting the fact that $k \in K_2$. Since $\rho^k \to 0$, we know that for all sufficiently large $k \in K_\epsilon$ we have $\rho^k \|\widehat{\lambda}^{k+1}\|_\infty < 1$, implying that ρ^k is not reduced by (3.2.18). Now consider (3.2.19). By (3.3.2f), we find that for $k \in K_\epsilon$ we obtain $\widehat{r}^k_{\mathcal{E}^k_c} = \widehat{s}^k_{\mathcal{E}^k_c} = 0$ and $\widehat{t}^k_{\mathcal{I}^k_c} = 0$. This implies that $\Delta l(\widehat{d}^k; x^k) = \Delta l(\overline{d}^k; x^k) = v(x^k)$, so we obtain $w^k = 0 < \omega$, $d^k = \widehat{d}^k$, and $\Delta l(d^k, x^k) = v(x^k)$. We then find that when the algorithm encounters (3.2.19), we have (temporarily using $\widehat{H}^k$ to denote $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$)

$$\begin{aligned}
\Delta m(d^k; x^k, \rho^k) - (d^k)^T \widehat{H}^k d^k &= -\rho^k \nabla f(x^k)^T d^k + \Delta l(d^k, x^k) - (d^k)^T \widehat{H}^k d^k \\
&= (d^k)^T \nabla c_{\mathcal{E}}(x^k) \widehat{\lambda}_{\mathcal{E}}^{k+1} + (d^k)^T \nabla c_{\mathcal{I}}(x^k) \widehat{\lambda}_{\mathcal{I}}^{k+1} + \Delta l(d^k; x^k) \\
&= \left(\|c_{\mathcal{E}}(x^k)\|_1 + \|[c_{\mathcal{I}}(x^k)]^+\|_1 \right) \\
&\qquad - c_{\mathcal{E}}(x^k)^T \widehat{\lambda}_{\mathcal{E}}^{k+1} - c_{\mathcal{I}}(x^k)^T \widehat{\lambda}_{\mathcal{I}}^{k+1} \\
&= \left(\|c_{\mathcal{E}}(x^k)\|_1 - c_{\mathcal{E}}(x^k)^T \widehat{\lambda}_{\mathcal{E}}^{k+1} \right) \\
&\qquad + \left(\|[c_{\mathcal{I}}(x^k)]^+\|_1 - c_{\mathcal{I}}(x^k)^T \widehat{\lambda}_{\mathcal{I}}^{k+1} \right). \qquad (3.3.13)
\end{aligned}$$

Here, the first equality follows by the definition of $\Delta m(d^k; x^k, \rho^k)$ and the second follows by (3.3.2a). Then, since (3.3.2g) implies $\widehat{\lambda}_{\mathcal{I}}^{k+1} \geq 0$, we find that for all $i \in \mathcal{I}$ either $\widehat{\lambda}_i^{k+1} = 0$ or, by (3.3.2e), $\widehat{\lambda}_i^{k+1} > 0$ and $\nabla c_i(x^k)^T \widehat{d}^k = -c_i(x^k)$. Consequently, we have

$$(d^k)^T \nabla c_{\mathcal{I}}(x^k) \widehat{\lambda}_{\mathcal{I}}^{k+1} = -c_{\mathcal{I}}(x^k)^T \widehat{\lambda}_{\mathcal{I}}^{k+1}. \qquad (3.3.14)$$

This along with (3.3.2b), (3.3.2c), and the definition of $\Delta l(d^k; x^k)$ yield the third and fourth equalities above, the latter being a rearrangement of the former. Since $\widehat{\lambda}_{\mathcal{E}}^{k+1} \leq \|\widehat{\lambda}_{\mathcal{E}}^{k+1}\|_\infty e$, we have $c_{\mathcal{E}}(x^k)^T \widehat{\lambda}_{\mathcal{E}}^{k+1} \leq \|\widehat{\lambda}_{\mathcal{E}}^{k+1}\|_\infty \|c_{\mathcal{E}}(x^k)\|_1$, and as $0 \leq \widehat{\lambda}_{\mathcal{I}}^{k+1} \leq \|\widehat{\lambda}_{\mathcal{I}}^{k+1}\|_\infty e$,

$$c_{\mathcal{I}}(x^k)^T \widehat{\lambda}_{\mathcal{I}}^{k+1} \leq [c_{\mathcal{I}}(x^k)]^{+T} \widehat{\lambda}_{\mathcal{I}}^{k+1} \leq \|[c_{\mathcal{I}}(x^k)]^+\|_1 \|\widehat{\lambda}^{k+1}\|_\infty.$$

Consequently, we have from (3.3.13) that

$$\begin{aligned}
\Delta m(d^k; x^k, \rho^k) \geq\ & (d^k)^T \widehat{H}^k d^k + (1 - \|\widehat{\lambda}^{k+1}\|_\infty) \|c_{\mathcal{E}}(x^k)\|_1 + (1 - \|\widehat{\lambda}^{k+1}\|_\infty) \|[c_{\mathcal{I}}(x^k)]^+\|_1 \\
=\ & (d^k)^T \widehat{H}^k d^k + (1 - \|\widehat{\lambda}^{k+1}\|_\infty) \Delta l(d^k; x^k) \\
\geq\ & \epsilon \Delta l(d^k; x^k), \qquad (3.3.15)
\end{aligned}$$

meaning that ρ^k will not be reduced by (3.2.19). Overall, we have contradicted the fact that $k \in K_2$. Hence, we have shown that for large $k \in K_2$, we have $\|\widehat{\lambda}^{k+1}\|_\infty > 1 - \epsilon$.

Now let $\widetilde{\rho}^k = \widehat{\rho}^k / \|\widehat{\lambda}^{k+1}\|$ and $\widetilde{\lambda}^{k+1} = \widehat{\lambda}^{k+1} / \|\widehat{\lambda}^{k+1}\|_\infty$ be defined for all $k \in K_2$ such that $\|\widehat{\lambda}^{k+1}\|_\infty > 1 - \epsilon$. Since there is an infinite number of such k, it follows that $\widetilde{\rho}^k \to 0$ and there exists a nonzero limit point $\widetilde{\lambda}^*$ of $\{\widetilde{\lambda}^{k+1}\}_{k \in K_2}$. Consider an infinite subsequence $K_\lambda \subseteq K_2$ such that $\{(x^k, \widetilde{\lambda}^{k+1})\}_{k \in K_\lambda} \to (x^*, \widetilde{\lambda}^*)$. By (3.3.2a), we find that for $k \in K_\lambda$,

$$\nabla_x \mathcal{F}(x^k, \widetilde{\rho}^k, \widetilde{\lambda}^{k+1}) = \widetilde{\rho}^k \nabla f(x^k)^T + \nabla c_{\mathcal{E}}(x^k) \widetilde{\lambda}_{\mathcal{E}}^{k+1} + \nabla c_{\mathcal{I}}(x^k) \widetilde{\lambda}_{\mathcal{I}}^{k+1} = \widehat{H}^k d^k / \|\widehat{\lambda}^{k+1}\|_\infty.$$

Since $d^k \to 0$ by Lemma 3.3.9 and $\|\widehat{\lambda}^{k+1}\|_\infty$ is bounded below for sufficient large $k \in K_\lambda$, we have that under Assumption (3.3.3)

$$\nabla_x \mathcal{F}(x^*, 0, \lambda^*) = \lim_{\substack{k \in K_\lambda \\ k \to \infty}} \nabla_x \mathcal{F}(x^k, \widetilde{\rho}^k, \widetilde{\lambda}^{k+1}) = 0.$$

Moreover, since $\|\widetilde{\lambda}^{k+1}\|_\infty$ is bounded, as in (3.3.12), we have

$$c_{\mathcal{I}}(x^*)^T \widetilde{\lambda}_{\mathcal{I}}^* = \lim_{\substack{k \in K_\lambda \\ k \to \infty}} c_{\mathcal{I}}(x^k)^T \widehat{\lambda}_{\mathcal{I}}^{k+1} / \|\widehat{\lambda}^{k+1}\|_\infty = 0.$$

Overall, we have shown that that $(x^*, 0, \widetilde{\lambda}^*)$ is a FJ point for problem (3.2.1).

Let $(x^*, 0, \lambda^*)$ be a FJ point as described above where $\lambda^* = \overline{\lambda}^*$ if we are in Case 1 and $\lambda^* = \widetilde{\lambda}^*$ if we are in Case 2. Then, from dual feasibility in (3.2.4) we have

$$\nabla_x \mathcal{F}(x^*, 0, \lambda^*) = \nabla c_{\mathcal{I}}(x^*) \lambda_{\mathcal{I}}^* + \nabla c_{\mathcal{E}}(x^*) \lambda_{\mathcal{E}}^* = 0. \tag{3.3.16}$$

Moreover, from the complementarity conditions in (3.2.4), we have

$$\nabla c_{\mathcal{A}^*}(x^*) \lambda_{\mathcal{A}^*}^* + \nabla c_{\mathcal{E}}(x^*) \lambda_{\mathcal{E}}^* = 0. \tag{3.3.17}$$

In order to derive a contradiction, suppose that the MFCQ holds at x^*. Since the MFCQ holds and $v(x^*) = 0$, there exists a vector u such that $\nabla c_{\mathcal{A}^*}(x^*)^T u < 0$ and $\nabla c_{\mathcal{E}}(x^*)^T u = 0$.

By (3.3.17), we then have

$$0 = u^T \nabla c_{\mathcal{A}^*}(x^*)\lambda^*_{\mathcal{A}^*} + u^T \nabla c_{\mathcal{E}}(x^*)\lambda^*_{\mathcal{E}} = u^T \nabla c_{\mathcal{A}^*}(x^*)\lambda^*_{\mathcal{A}^*}. \tag{3.3.18}$$

Since $\nabla c_{\mathcal{A}^*}(x^*)^T u < 0$ and $\lambda^*_{\mathcal{A}^*} \geq 0$, (3.3.18) implies $\lambda^*_{\mathcal{A}^*} = 0$. Thus, from (3.3.17) and the fact that under the MFCQ the columns of $\nabla c_{\mathcal{E}}(x^*)$ are linearly independent, we have $\lambda^*_{\mathcal{E}} = 0$. Overall, we have shown that $\lambda^* = 0$, which contradicts the fact that $(x^*, 0, \lambda^*)$ is a FJ point for problem (3.2.1). Hence, MFCQ fails at x^*. □

We end our global convergence theory with a corollary that summarizes the results of the previous theorems. It also provides a stronger result in a special case when the primal iterates are bounded. This occurs, e.g., when the sublevel sets of the shifted penalty function $\varphi(\cdot, \rho)$ (recall (3.3.5)) are bounded for all ρ in the closure of $\{\rho^k\}$.

Corollary 3.3.7. *Suppose Assumption 3.3.3 holds and let K_ρ be defined as in Theorem 3.3.6. Then, one of the following situations occurs:*

(i) $\rho^k \to \rho^*$ *for some constant* $\rho^* > 0$ *and each limit point of* $\{x^k\}$ *either corresponds to a KKT point or an infeasible stationary point for problem* (3.2.1)*;*

(ii) $\rho^k \to 0$ *and all limit points of* $\{x^k\}$ *are infeasible stationary points for* (3.2.1)*;*

(iii) $\rho^k \to 0$*, all limit points of* $\{x^k\}$ *are feasible for* (3.2.1)*, and all limit points of* $\{x^k\}_{k \in K_\rho}$ *correspond to FJ points for* (3.2.1) *where the MFCQ fails.*

Consequently, if $\{x^k\}$ *is bounded and all limit points of this sequence are feasible for* (3.2.1) *and satisfy the MFCQ, then* $\rho^k \to \rho^*$ *for some constant* $\rho^* > 0$ *and all limit points of* $\{x^k\}$ *are KKT points for problem* (3.2.1).

Proof. The fact that one of situations (i)–(iii) occurs follows from the Theorems 3.3.4–3.3.6 and the fact that $\{\rho^k\}$ is monotonically decreasing and bounded below by zero. All that remains is to prove the last sentence of the corollary. In order to derive a contradiction, suppose that under the stated conditions we have $\rho^k \to 0$. Then, since $\{x^k\}$ is bounded, it follows that the sequence $\{x^k\}_{k \in K_\rho}$ has at least one limit point. However, by Theorem 3.3.6, it follows that such a limit point violates the MFCQ, which in turn contradicts the stated conditions. Hence, $\rho^k \to \rho^*$ for some constant $\rho^* > 0$ and $v(x^k) \to 0$, so the result follows from Theorem 3.3.5. $\square$

3.3.3 Local Convergence

We consider the local convergence of `SQuID` in the neighborhood of first-order optimal points satisfying certain common assumptions, delineated below. For the most part, our assumptions in this subsection represent a strengthening of the assumptions in §3.3.2. However, we loosen our assumptions on the quadratic terms in subproblems (3.2.7) and (3.2.9) as in this subsection we only require that they are positive definite in the null space of the Jacobian of the constraints that are active at a given first-order optimal point.

First, for a given x^*, we will use the following assumption.

Assumption 3.3.8. *The problem functions f, $c_{\mathcal{E}}$, and $c_{\mathcal{I}}$ and their first and second derivatives are bounded and Lipschitz continuous in an open convex set containing x^*.*

Second, we make the following assumption concerning a given stationary point x^* of (3.2.2). As such a point may be feasible or infeasible for (3.2.1), we make this assumption throughout our local analysis.

Assumption 3.3.9. *Let x^* be a first-order optimal point for* (3.2.2) *such that there exists $\overline{\lambda}^*$ with $(x^*, \overline{\lambda}^*)$ satisfying* (3.2.5). *Then, Assumption 3.3.8 holds at x^* and*

(a) $\nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^*)^T$ *has full row rank;*

(b) $-e < \overline{\lambda}^*_{\mathcal{Z}^*} < e$ *and* $0 < \overline{\lambda}^*_{\mathcal{A}^*} < e$*;*

(c) $d^T H(x^*, 0, \overline{\lambda}^*) d > 0$ *for all* $d \neq 0$ *such that* $\nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^*)^T d = 0$.

Moreover, the following hold true for the iterates generated by `SQuID` *:*

(d) $x^k \to x^*$.

(e) *For all large* k, $H(x^k, 0, \overline{\lambda}^k)$ *and* $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$ *are the exact Hessian of* $\mathcal{F}$ *at* $(x^k, 0, \overline{\lambda}^k)$ *and* $(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$, *respectively.*

(f) *For all large* k, $\alpha^k = 1$.

Finally, if $x^k \to x^*$, where x^* is a KKT point for (3.2.1), we make the following assumption. (While we state Assumption 3.3.10 now, we will not use it until §3.3.3.)

Assumption 3.3.10. *Let* x^* *be a first-order optimal point for* (3.2.1) *such that Assumption 3.3.9 holds and there are* $\rho^* > 0$ *and* $\widehat{\lambda}^*$ *with* $(x^*, \rho^*, \widehat{\lambda}^*)$ *satisfying* (3.2.4). *Then,*

(a) $\rho^k \to \rho^*$;

(b) $\widehat{\lambda}^*_{\mathcal{A}^*} + c_{\mathcal{A}^*}(x^*) > 0$;

(c) $d^T H(x^*, \rho^*, \widehat{\lambda}^*) d > 0$ *for all* $d \neq 0$ *such that* $\nabla c_{\mathcal{E}^* \cup \mathcal{A}^*}(x^*)^T d = 0$.

The assumptions above may be viewed as strong when one considers the fact that local superlinear convergence guarantees for SQO methods have been provided in more general settings. Our algorithm is able to achieve such convergence in such settings, but accounting for more general conditions would only add unnecessary complications to the analysis and detract attention away from our central focus, i.e., the novel feature of attaining superlinear convergence for both feasible and infeasible problem instances with a single algorithm. In particular, consider Assumptions 3.3.9(e) and (f). The former of these assumptions is strong since, if an exact Hessian is indefinite, the algorithm must ensure that of all of the local minimizers of the corresponding QO subproblem, the subproblem solver computes one satisfying certain conditions (implicit in Lemma 3.23 later on). This is challenging as nonconvex quadratic optimization is known to be NP-hard [66]. On the other hand, assuming only that the Hessian is positive definite in the null space of the active constraint Jacobian, the algorithm could ensure that the QO subproblem has a unique solution by modifying the Hessian in appropriate ways so that fast local convergence is still possible. For example, this can be achieved by augmenting the Hessian with

$\sigma \nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k) \nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k)^T$ for a sufficiently large $\sigma > 0$ and then applying the characterization result for superlinear convergence found in [3]. As for Assumption 3.3.9(f), the primary practical concern is the Maratos effect [60], which makes this assumption inappropriate in many cases. However, we may assume that a watchdog mechanism [22] or a second-order correction [35] is employed to ensure that unit steplengths are accepted by the line search for large k. We leave it a subject of future research to see how many of the assumptions above (in addition to Assumptions 3.3.9(e) and (f)) can be relaxed while still ensuring the convergence guarantees below, potentially with minor algorithmic variations.

Local convergence to an infeasible stationary point

Suppose Assumption 3.3.9 holds where x^* is an infeasible stationary point for (3.2.1). We show that, in such cases, `SQuID` converges quadratically to $(x^*, \overline{\lambda}^*)$. Some of our analysis for this case follows that in [13], though we provide proofs for completeness.

A critical component of our local convergence analysis in this subsection is to show that there is an inherent relationship between problem (3.2.2) and the following:

$$\min_{(x, r_{\mathcal{P}^*}, s_{\mathcal{N}^*}, t_{\mathcal{V}^*})} \rho f(x) + e^T r_{\mathcal{P}^*} + e^T s_{\mathcal{N}^*} + e^T t_{\mathcal{V}^*}$$
$$\text{s.t.} \begin{cases} c_{\mathcal{P}^*}(x) = r_{\mathcal{P}^*}, \quad c_{\mathcal{Z}^*}(x) = 0, \quad -c_{\mathcal{N}^*}(x) = s_{\mathcal{N}^*}, \\ c_{\mathcal{V}^*}(x) \leq t_{\mathcal{V}^*}, \quad c_{\mathcal{A}^* \cup \mathcal{S}^*}(x) \leq 0, \\ (r_{\mathcal{P}^*}, s_{\mathcal{N}^*}, t_{\mathcal{V}^*}) \geq 0. \end{cases} \tag{3.3.19}$$

In particular, in our first two lemmas, we establish that solutions of (3.3.19) converge to that of (3.2.2) as $\rho \to 0$.

The following lemma shows that x^* corresponds to a solution of (3.3.19) for $\rho = 0$.

Lemma 3.3.10. *Suppose Assumption 3.3.9 holds and $v(x^*) > 0$. Then, x^* and*

$$(r^*_{\mathcal{P}^*}, s^*_{\mathcal{N}^*}, t^*_{\mathcal{V}^*}) = (c_{\mathcal{P}^*}(x^*), -c_{\mathcal{N}^*}(x^*), c_{\mathcal{V}^*}(x^*))$$

correspond to a first-order optimal point for (3.3.19) *for $\rho = 0$. Moreover, the correspond-*

ing dual solution is the unique $\overline{\lambda}^*$ *such that* $(x^*, \overline{\lambda}^*)$ *satisfies* (3.2.5).

Proof. First-order optimality conditions for (3.3.19) are the following:

$$\rho \nabla f(x) + \nabla c_{\mathcal{E}}(x)\lambda_{\mathcal{E}} + \nabla c_{\mathcal{I}}(x)\lambda_{\mathcal{I}} = 0, \tag{3.3.20a}$$

$$c_{\mathcal{P}^*}(x) = r_{\mathcal{P}^*}, \quad c_{\mathcal{Z}^*}(x) = 0, \quad -c_{\mathcal{N}^*}(x) = s_{\mathcal{N}^*}, \tag{3.3.20b}$$

$$c_{\mathcal{V}^*}(x) \le 0, \quad c_{\mathcal{A}^* \cup \mathcal{S}^*}(x) \le 0, \tag{3.3.20c}$$

$$(r_{\mathcal{P}^*}, s_{\mathcal{N}^*}, t_{\mathcal{V}^*}) \ge 0, \tag{3.3.20d}$$

$$\lambda_{\mathcal{A}^* \cup \mathcal{S}^*} \cdot c_{\mathcal{A}^* \cup \mathcal{S}^*}(x) = 0, \tag{3.3.20e}$$

$$\lambda_{\mathcal{V}^*} \cdot (c_{\mathcal{V}^*}(x) - t_{\mathcal{V}^*}) = 0, \tag{3.3.20f}$$

$$(e - \lambda_{\mathcal{P}^*}) \cdot r_{\mathcal{P}^*} = 0, \quad (e + \lambda_{\mathcal{N}^*}) \cdot s_{\mathcal{N}^*} = 0, \quad (e - \lambda_{\mathcal{V}^*}) \cdot t_{\mathcal{V}^*} = 0, \tag{3.3.20g}$$

$$\lambda_{\mathcal{P}^*} \le e, \ \lambda_{\mathcal{N}^*} \ge -e, \ \lambda_{\mathcal{A}^* \cup \mathcal{S}^*} \ge 0, \ 0 \le \lambda_{\mathcal{V}^*} \le 0. \tag{3.3.20h}$$

If x^* is an infeasible stationary point, then by definition there exists $\overline{\lambda}^* \neq 0$ such that $(x^*, \overline{\lambda}^*)$ satisfies (3.2.5). Then, with $r_{\mathcal{P}^*}$, $s_{\mathcal{N}^*}$, and $t_{\mathcal{V}^*}$ chosen as in the statement of the lemma, it is easily verified that $(x^*, r_{\mathcal{P}^*}, s_{\mathcal{N}^*}, t_{\mathcal{V}^*}, \overline{\lambda}^*)$ satisfies (3.3.20) for $\rho = 0$. Moreover, from (3.3.20e) and (3.3.20g), we find $\overline{\lambda}^*_{\mathcal{S}^*} = 0$, $\overline{\lambda}^*_{\mathcal{P}^*} = e$, $\overline{\lambda}^*_{\mathcal{N}^*} = -e$, and $\overline{\lambda}_{\mathcal{V}^*} = e$. These equations and (3.3.20a) imply that we have

$$\nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^*)\overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*} = -\nabla c_{\mathcal{P}^* \cup \mathcal{V}^*}(x^*)e + \nabla c_{\mathcal{N}^*}(x^*)e. \tag{3.3.21}$$

Under Assumption 3.3.9(a), $\overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*}$ in (3.3.21) is unique. Thus, $\overline{\lambda}^*$ is unique. □

We now show that for sufficiently small $\rho > 0$, the solution of problem (3.3.19) shares critical properties with that of problem (3.2.2). This result is formalized in our next lemma, which makes use of the following nonlinear system of equations:

$$\begin{aligned} 0 &= F(x, \rho, \lambda_{\mathcal{Z}^* \cup \mathcal{A}^*}) \\ &:= \begin{bmatrix} \rho \nabla f(x) + \nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x)\lambda_{\mathcal{Z}^* \cup \mathcal{A}^*} + \nabla c_{\mathcal{P}^* \cup \mathcal{V}^*}(x)e - \nabla c_{\mathcal{N}^*}(x)e \\ c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x) \end{bmatrix}. \end{aligned} \tag{3.3.22}$$

By differentiating F with respect to $(x, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*})$, we obtain

$$F'(x, \rho, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) := \frac{\partial F(x, \rho, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*})}{\partial(x, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*})} = \begin{bmatrix} G(x, \rho, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) & \nabla c_{\mathcal{Z}^*\cup\mathcal{A}^*}(x) \\ \nabla c_{\mathcal{Z}^*\cup\mathcal{A}^*}(x)^T & 0 \end{bmatrix} \tag{3.3.23}$$

where

$$G(x, \rho, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) := \rho\nabla_2 f(x) + \sum_{i\in\mathcal{P}^*\cup\mathcal{V}^*} \nabla_2 c_i(x) + \sum_{i\in\mathcal{Z}^*\cup\mathcal{A}^*} \lambda_i \nabla_2 c_i(x) - \sum_{i\in\mathcal{N}^*} \nabla_2 c_i(x).$$

Lemma 3.3.11. *Suppose Assumption 3.3.9 holds and $v(x^*) > 0$. Then, for all ρ sufficiently small, problem* (3.3.19) *has a solution $(x^\rho, r^\rho_{\mathcal{P}^*}, s^\rho_{\mathcal{N}^*}, t^\rho_{\mathcal{V}^*})$ where x^ρ yields the same sets of positive, zero, and negative-valued equality constraints and violated, active, and strictly satisfied inequality constraints as x^*. Moreover, for such ρ, the corresponding dual variables satisfy $\overline{\lambda}^\rho_{\mathcal{P}^*} = e$, $-e < \overline{\lambda}^\rho_{\mathcal{Z}^*} < e$, $\overline{\lambda}^\rho_{\mathcal{N}^*} = -e$, $\overline{\lambda}^\rho_{\mathcal{V}^*} = e$, $0 < \overline{\lambda}^\rho_{\mathcal{A}^*} < e$, and $\overline{\lambda}^\rho_{\mathcal{S}^*} = 0$, and we have*

$$\left\| \begin{bmatrix} x^\rho - x^* \\ \overline{\lambda}^\rho - \overline{\lambda}^* \end{bmatrix} \right\| = O(\rho). \tag{3.3.24}$$

Proof. Under Assumption 3.3.8, F in (3.3.22) is a continuously differentiable mapping about $(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*})$, and under Assumption 3.3.9(a) and (c), the matrix F' in (3.3.23) is nonsingular at $(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*})$. Thus, by the implicit function theorem [75, Theorem 9.28], there exist open sets $\mathcal{B}_x \subset \mathbb{R}^n$, $\mathcal{B}_\rho \subset \mathbb{R}$, and $\mathcal{B}_\lambda \subset \mathbb{R}_{|\mathcal{Z}^*\cup\mathcal{A}^*|}$ containing x^*, 0, and $\overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*}$, respectively, and continuously differentiable functions $x(\rho) : \mathcal{B}_\rho \to \mathcal{B}_x$ and $\overline{\lambda}_{\mathcal{Z}^*\cup\mathcal{A}^*}(\rho) : \mathcal{B}_\rho \to \mathcal{B}_\lambda$ such that

$$x(0) = x^*, \quad \overline{\lambda}_{\mathcal{Z}^*\cup\mathcal{A}^*}(0) = \overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*}, \quad \text{and} \quad F(x(\rho), \rho, \overline{\lambda}_{\mathcal{Z}^*\cup\mathcal{A}^*}(\rho)) = 0 \;\; \text{for all} \;\; \rho \in \mathcal{B}_\rho.$$

By the second equation in (3.3.22) and since $x(\rho)$ varies continuously with ρ, we have

$$c_{\mathcal{Z}^*\cup\mathcal{A}^*}(x(\rho)) = 0, \quad c_{\mathcal{P}^*\cup\mathcal{V}^*}(x(\rho)) > 0, \quad \text{and} \quad c_{\mathcal{N}^*\cup\mathcal{S}^*}(x(\rho)) < 0 \tag{3.3.25}$$

for ρ sufficiently small. Similarly, since $-e < \overline{\lambda}^*_{\mathcal{Z}^*} < e$ and $0 < \overline{\lambda}^*_{\mathcal{A}^*} < e$ under Assumption 3.3.9(b), the fact that $\overline{\lambda}_{\mathcal{Z}^* \cup \mathcal{A}^*}(\rho)$ varies continuously with ρ implies that $-e < \overline{\lambda}_{\mathcal{Z}^*}(\rho) < e$ and $0 < \overline{\lambda}_{\mathcal{A}^*}(\rho) < e$ for ρ sufficiently small. If we define

$$\overline{\lambda}_{\mathcal{P}^* \cup \mathcal{V}^*}(\rho) := e, \quad \overline{\lambda}_{\mathcal{N}^*}(\rho) := -e, \quad \text{and} \quad \overline{\lambda}_{\mathcal{S}^*}(\rho) := 0$$

along with

$$r_{\mathcal{P}_*}(\rho) := [c_{\mathcal{P}^*}(x(\rho))]^+, \quad s_{\mathcal{N}^*}(\rho) := [c_{\mathcal{N}^*}(x(\rho))]^-, \quad \text{and} \quad t_{\mathcal{V}^*}(\rho) := [c_{\mathcal{V}^*}(x(\rho))]^+,$$

it follows that $(x(\rho), r_{\mathcal{P}^*}(\rho), s_{\mathcal{N}^*}(\rho), t_{\mathcal{V}^*}(\rho), \overline{\lambda}(\rho))$ satisfies (3.3.20), and is therefore a first-order optimal point for (3.3.19) for sufficiently small ρ. Hence, by (3.3.25), we have that $x^\rho = x(\rho)$ for ρ sufficiently small has the same sets of positive, zero, and negative-valued equality and violated, active, and strictly satisfied inequality constraints as x^*.

All that remains is to establish (3.3.24). From the differentiability of $x^\rho = x(\rho)$ and $\overline{\lambda}^\rho_{\mathcal{Z}^* \cup \mathcal{A}^*} = \overline{\lambda}_{\mathcal{Z}^* \cup \mathcal{A}^*}(\rho)$ and their derivatives given by the implicit function theorem, we have for ρ sufficiently small that

$$\begin{bmatrix} x^\rho \\ \overline{\lambda}^\rho_{\mathcal{Z}^* \cup \mathcal{A}^*} \end{bmatrix} = \begin{bmatrix} x^* \\ \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*} \end{bmatrix} - F'_{x, \lambda_{\mathcal{Z}^* \cup \mathcal{A}^*}}(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})_{-1} F'_\rho(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})\rho + o(\rho).$$

Hence, under Assumption 3.3.9, (3.3.24) is satisfied. □

We now turn back to the iterates produced by `SQuID` . In particular, as in the previous lemma, we show that in a neighborhood of an infeasible stationary point, subproblems (3.2.7) and (3.2.9) will suggest the optimal partition of the index sets $\mathcal{E}$ and $\mathcal{I}$. This result is reminiscent of the well-known result in [70].

Lemma 3.3.12. *Suppose Assumption 3.3.9 holds and $v(x^*) > 0$. Then, for all $\widehat{\rho}^k$ sufficiently small and for all $(x^k, \overline{\lambda}^k)$ and $(x^k, \widehat{\lambda}^k)$ each sufficiently close to $(x^*, \overline{\lambda}^*)$:*

(a) There are local solutions for (3.2.7) *and* (3.2.9) *such that $\overline{d}^k$ and $\widehat{d}^k$ yield the same*

sets of positive, zero, and negative-valued equality and violated, active, and strictly satisfied inequality constraints as x^*. *Moreover, with* $(\rho, H) = \left(0, H(x^k, 0, \overline{\lambda}^k)\right)$ *and* $(\rho, H) = \left(\widehat{\rho}^k, H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\right)$, *respectively, the optimal solutions for* (3.2.7) *and* (3.2.9) *satisfy*

$$\begin{bmatrix} H & \nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k) \\ \nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda_{\mathcal{Z}^* \cup \mathcal{A}^*} \end{bmatrix} = - \begin{bmatrix} \rho \nabla f(x^k) + \nabla c_{\mathcal{P}^* \cup \mathcal{V}^*}(x^k) - \nabla c_{\mathcal{N}^*}(x^k) \\ c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k) \end{bmatrix} \quad (3.3.26)$$

and

$$\lambda_{\mathcal{P}^* \cup \mathcal{V}^*} = e, \ -e < \lambda_{\mathcal{Z}^*} < e, \ \lambda_{\mathcal{N}^*} = -e, \ 0 < \lambda_{\mathcal{A}^*} < e, \ \textit{and } \lambda_{\mathcal{S}^*} = 0. \quad (3.3.27)$$

(b) The update (3.2.13) *is triggered infinitely often, yielding* $(\rho^k, \widehat{\rho}^k) \to 0$.

Proof. For part (a), consider subproblem (3.2.7), meaning that we let $(\rho, H) = (0, H(x^k, 0, \overline{\lambda}^k))$ in (3.3.26). With $\overline{d}^k = 0$, (3.3.10) reduces to (3.2.5). Thus, (3.3.10) is solved at $(x^*, \overline{\lambda}^*)$ by $(d, \lambda) = (0, \overline{\lambda}^*)$. By (3.3.10c)–(3.3.10f), we have $\overline{\lambda}^*_{\mathcal{P}^* \cup \mathcal{V}^*} = e$, $\overline{\lambda}^*_{\mathcal{N}^*} = -e$, and $\overline{\lambda}^*_{\mathcal{S}^*} = 0$. Hence, by (3.3.10a) and the definitions of $\mathcal{Z}^*$ and $\mathcal{A}^*$, the linear system (3.3.26) is satisfied at $(x^*, \overline{\lambda}^*)$ by $(d, \lambda_{\mathcal{Z}^* \cup \mathcal{A}^*}) = (0, \overline{\lambda}_{\mathcal{Z}^* \cup \mathcal{A}^*})$. Under Assumption 3.3.9(a) and (c), the matrix in (3.3.26) is nonsingular at $(x^*, \overline{\lambda}^*)$, and hence the solution of (3.3.26) varies continuously in a neighborhood of $(x^*, \overline{\lambda}^*)$. In addition, under Assumption 3.3.9(c), it follows that $H = H(x^k, 0, \overline{\lambda}^k)$ in (3.3.26) is positive definite on the null space of $\nabla c_{\mathcal{Z}^* \cup \mathcal{A}^*}(x^k)^T$ in a neighborhood of $(x^*, \overline{\lambda}^*)$.

It follows from the conclusions in the previous paragraph that for all $(x^k, \overline{\lambda}^k)$ sufficiently close to $(x^*, \overline{\lambda}^*)$, the solution $(\overline{d}^k, \overline{\lambda}^{k+1}_{\mathcal{Z}^* \cup \mathcal{A}^*})$ to (3.3.26) is sufficiently close to $(0, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})$

such that it satisfies

$$-e < \overline{\lambda}^{k+1}_{\mathcal{Z}^*} < e, \quad 0 < \overline{\lambda}^{k+1}_{\mathcal{A}^*} < e,$$

$$c_{\mathcal{P}^*\cup\mathcal{V}^*}(x^k) + \nabla c_{\mathcal{P}^*\cup\mathcal{V}^*}(x^k)^T \overline{d}^k > 0,$$

$$\text{and} \quad c_{\mathcal{N}^*\cup\mathcal{S}^*}(x^k) + \nabla c_{\mathcal{N}^*\cup\mathcal{S}^*}(x^k)^T \overline{d}^k < 0.$$

By construction, such a solution $(\overline{d}^k, \overline{\lambda}^{k+1}_{\mathcal{Z}^*\cup\mathcal{A}^*})$ satisfies (3.3.26) and therefore satisfies (3.3.10) together with $\overline{\lambda}^{k+1}_{\mathcal{P}^*\cup\mathcal{V}^*} = e$, $\overline{\lambda}^{k+1}_{\mathcal{N}^*} = -e$, and $\overline{\lambda}^{k+1}_{\mathcal{S}^*} = 0$. Therefore, $(\overline{d}^k, \overline{\lambda}^{k+1})$ is a KKT point of subproblem (3.2.7), and, as revealed above, it identifies the same sets of positive, zero, and negative-valued equality and violated, active, and strictly satisfied inequality constraints as x^*.

The proof of the result corresponding to subproblem (3.2.9) is similar. Indeed, from the discussion above, we find that for ρ^k (and hence $\widehat{\rho}^k$) sufficiently small and $(x^k, \overline{\lambda}^k)$ sufficiently close to $(x^*, \overline{\lambda}^*)$, the algorithm will set $\mathcal{E}^k = \mathcal{Z}^*$, $\mathcal{E}^k_c = \mathcal{P}^* \cup \mathcal{N}^*$, $\mathcal{I}^k = \mathcal{A}^* \cup \mathcal{S}^*$ and $\mathcal{I}^k_c = \mathcal{V}^*$. The remainder of the proof follows as above with $H(x^k, 0, \overline{\lambda}^k)$, (3.3.10), and $(\overline{d}^k, \overline{\lambda}^{k+1})$ replaced by $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$, (3.3.2), and $(\widehat{d}^k, \widehat{\lambda}^{k+1})$, respectively.

Now we prove part (b). We first argue that (3.2.12) holds for all sufficiently large k so that ρ^k is set by (3.2.13) infinitely many times. Then, we show that this yields $\rho^k \to 0$. As x^k approaches x^*, we have that $v(x^k) > \frac{1}{2}v(x^*) > 0$ for all large k. On the other hand, in a neighborhood of x^*, the constraint functions $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$ are bounded under Assumption 3.3.8. Thus, by the definition of $\Delta l(\overline{d}^k; x^k)$ and since for all $(x^k, \overline{\lambda}^k)$ sufficiently close to $(x^*, \overline{\lambda}^*)$, the solution $(\overline{d}^k, \overline{\lambda}^{k+1})$ to (3.3.26) is sufficiently close to $(0, \overline{\lambda}^*)$, we have that $\Delta l(\overline{d}^k; x^k) \leq \frac{\theta}{2}v(x^*) < \theta v(x^k)$ for sufficiently large k. Overall, this implies that (3.2.12) holds for such k. Hence, (3.2.13) is triggered infinitely many times. Finally, to see that (3.2.13) drives $\rho^k \to 0$, it suffices to see that $(\overline{d}^k, \overline{\lambda}^{k+1}) \to (0, \overline{\lambda}^*)$, (3.3.10a), and (3.3.10c)-(3.3.10f) yield $\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) \to 0$. □

Lemma 3.3.12 can be used to show that near $(x^*, \overline{\lambda}^*)$, the solutions of system (3.3.26) with $(\rho, H) = \left(0, H(x^k, 0, \overline{\lambda}^k)\right)$ and $(\rho, H) = \left(\widehat{\rho}^k, H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)\right)$ correspond to Newton steps for $F(x, 0, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) = 0$ and $F(x, \widehat{\rho}^k, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) = 0$, respectively. We formalize this property in the following lemma.

Lemma 3.3.13. *Suppose Assumption 3.3.9 holds and $v(x^*) > 0$. Then:*

(a) If $(x^k, \overline{\lambda}^k)$ is sufficiently close to $(x^*, \overline{\lambda}^*)$ and $(\overline{d}^k, \overline{\lambda}^{k+1})$ generated by subproblem (3.2.7) is obtained via (3.3.26) with $(\rho, H) = (0, H(x^k, 0, \overline{\lambda}^k))$, then

$$\left\| \begin{bmatrix} x^k + \overline{d}^k - x^* \\ \overline{\lambda}^{k+1} - \overline{\lambda}^* \end{bmatrix} \right\| \leq \overline{C} \left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\|_2 \tag{3.3.28}$$

for some constant $\overline{C} > 0$ independent of k.

(b) If $(x^k, \widehat{\lambda}^k)$ is sufficiently close to $(x^*, \overline{\lambda}^*)$ and $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ generated by subproblem (3.2.9) is obtained via (3.3.26) with $(\rho, H) = (\widehat{\rho}^k, H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k))$, then, with (x^ρ, λ^ρ) defined as in Lemma 3.3.11, we have

$$\left\| \begin{bmatrix} x^k + \widehat{d}^k - x^\rho \\ \widehat{\lambda}^{k+1} - \lambda^\rho \end{bmatrix} \right\| \leq \widehat{C} \left\| \begin{bmatrix} x^k - x^\rho \\ \widehat{\lambda}^k - \lambda^\rho \end{bmatrix} \right\|_2 \tag{3.3.29}$$

for some constant $\widehat{C} > 0$ independent of k.

Proof. For both parts (a) and (b), our proof follows that of [13, Lemma 3.5].

For part (a), by Lemma 3.3.12(a), if $(x^k, \overline{\lambda}^k)$ is sufficiently close to $(x^*, \overline{\lambda}^*)$, then $(\overline{d}^k, \overline{\lambda}^{k+1})$ generated by subproblem (3.2.7) can be obtained via (3.3.27) and (3.3.26) with $(\rho, H) = (0, H(x^k, 0, \overline{\lambda}^k))$. Therefore, since $H(x^k, 0, \overline{\lambda}^k) = G(x^k, 0, \overline{\lambda}^k_{\mathcal{Z}^*\cup\mathcal{A}^*})$ in such cases, (3.3.26) constitutes a Newton iteration for $F(x, 0, \lambda_{\mathcal{Z}^*\cup\mathcal{A}^*}) = 0$ at $(x^k, 0, \overline{\lambda}^k)$. We can now apply standard Newton analysis. By Assumption 3.3.8 we have that F is continuously differentiable and F' is Lipschitz continuous in a neighborhood of $(x^*, 0, \overline{\lambda}^*)$. By Assumption 3.3.9(a) and (c), the matrix F' is nonsingular at $(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*})$, so its inverse exists and is bounded in a neighborhood of $(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^*\cup\mathcal{A}^*})$. By [28, Theorem 5.2.1], if $(x^k, \overline{\lambda}^k)$ is sufficiently close to $(x^*, \overline{\lambda}^*)$, then we have that (3.3.28) holds true.

For part (b), by Lemma 3.3.12(a), if $(x^k, \widehat{\lambda}^k)$ is sufficiently close to $(x^*, \overline{\lambda}^*)$, then $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ generated by subproblem (3.2.7) can be obtained via (3.3.27) and (3.3.26) with $(\rho, H) = (\widehat{\rho}^k, H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k))$. Therefore, since $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k) = G(x^k, \widehat{\rho}^k, \widehat{\lambda}^k_{\mathcal{Z}^* \cup \mathcal{A}^*})$ in such cases, system (3.3.26) constitutes a Newton iteration for $F(x, \rho, \lambda_{\mathcal{Z}^* \cup \mathcal{A}^*}) = 0$ at $(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$. By Assumption 3.3.8 we have that F is continuously differentiable and F' is Lipschitz continuous in a neighborhood of $(x^*, \rho, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})$. Moreover, since ρ is bounded, the Lipschitz constant κ_1 for F' in a neighborhood of $(x^*, \rho, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})$ is independent of ρ. By Assumption 3.3.9(a) and (c), the matrix F' is nonsingular at $(x^*, 0, \overline{\lambda}^*_{\mathcal{Z}^* \cup \mathcal{A}^*})$, and hence its inverse exists and is bounded in norm by a constant κ_2 in a neighborhood of that point. By [28, Theorem 5.2.1],

$$\text{if} \quad \left\| \begin{bmatrix} x^k - x^\rho \\ \widehat{\lambda}^k - \lambda^\rho \end{bmatrix} \right\| \leq \frac{1}{\kappa_1 \kappa_2} \quad \text{then} \quad \left\| \begin{bmatrix} x^k + \widehat{d}^k - x^\rho \\ \widehat{\lambda}^{k+1} - \lambda^\rho \end{bmatrix} \right\| \leq \kappa_1 \kappa_2 \left\| \begin{bmatrix} x^k - x^\rho \\ \widehat{\lambda}^k - \lambda^\rho \end{bmatrix} \right\|_2 .$$

This can be achieved if ρ is sufficiently small such that (x^ρ, λ^ρ) and $(x^k, \widehat{\lambda}^k)$ satisfy

$$\left\| \begin{bmatrix} x^\rho - x^* \\ \lambda^\rho - \overline{\lambda}^* \end{bmatrix} \right\| \leq \frac{1}{4\kappa_1 \kappa_2} \quad \text{and} \quad \left\| \begin{bmatrix} x^k - x^* \\ \widehat{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\| \leq \frac{1}{4\kappa_1 \kappa_2}.$$

□

We are now ready to prove our main theorem concerning the local convergence of `SQuID` in the neighborhood of infeasible stationary points. The theorem shows that the convergence rate is dependent on how fast ρ is decreased and $\widehat{\lambda}^k$ approaches $\overline{\lambda}^k$.

Theorem 3.3.11. *Suppose Assumption 3.3.9 holds and* $v(x^*) > 0$*. Then, if* $(x^k, \overline{\lambda}^k)$ *and* $(x^k, \widehat{\lambda}^k)$ *are each sufficiently close to* $(x^*, \overline{\lambda}^*)$*,* $(\overline{d}^k, \overline{\lambda}^{k+1})$ *is obtained via* (3.3.27) *and* (3.3.26) *with* $(\rho, H) = (0, H(x^k, 0, \overline{\lambda}^k))$*, and* $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ *is obtained via* (3.3.27) *and* (3.3.26) *with* $(\rho, H) = (\widehat{\rho}^k, H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k))$*, then*

$$\left\| \begin{bmatrix} x^{k+1} - x^* \\ \overline{\lambda}^{k+1} - \overline{\lambda}^* \end{bmatrix} \right\| \leq C \left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\|_2 + O(\|\widehat{\lambda}^k - \overline{\lambda}^k\|) + O(\rho) \tag{3.3.30}$$

for some constant $C > 0$ *independent of* k*. Consequently, as* (3.2.13) *and* (3.2.14) *yield*

$$\rho^k = O\left(\left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\|_2 \right) \quad \textit{and} \quad \|\widehat{\lambda}^k - \overline{\lambda}^k\| = O\left(\left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\|_2 \right),$$

$\{(x^k, \overline{\lambda}^k)\}$ *converges to* $(x^*, \overline{\lambda}^*)$ *quadratically. If* (3.2.13) *and* (3.2.14) *merely yielded*

$$\rho^k = o\left(\left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\| \right) \quad \textit{and} \quad \|\widehat{\lambda}^k - \overline{\lambda}^k\| = o\left(\left\| \begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix} \right\| \right),$$

then $\{(x^k, \overline{\lambda}^k)\}$ *would converge to* $(x^*, \overline{\lambda}^*)$ *superlinearly.*

Proof. For a given $\rho > 0$, let (x^ρ, λ^ρ) be defined as in Lemma 3.3.11. Under Assumption 3.3.9(f), $x^{k+1} = x^k + w^k\overline{d}^k + (1-w^k)\widehat{d}^k$ for all k. Thus,

$$\begin{aligned}
\left\|\begin{bmatrix} x^{k+1}-x^* \\ \overline{\lambda}^{k+1}-\overline{\lambda}^* \end{bmatrix}\right\| &\leq \; w^k \left\|\begin{bmatrix} x^k+\overline{d}^k-x^* \\ \overline{\lambda}^{k+1}-\overline{\lambda}^* \end{bmatrix}\right\| + (1-w^k) \left\|\begin{bmatrix} x^k+\widehat{d}^k-x^* \\ \overline{\lambda}^{k+1}-\overline{\lambda}^* \end{bmatrix}\right\| \\
&\leq \; w^k\overline{C} \left\|\begin{bmatrix} x^k-x^* \\ \overline{\lambda}^k-\overline{\lambda}^* \end{bmatrix}\right\|_2 \\
&\qquad + (1-w^k)\left(\left\|\begin{bmatrix} x^k+\widehat{d}^k-x^\rho+x^\rho-x^* \\ 0 \end{bmatrix}\right\| + \left\|\begin{bmatrix} 0 \\ \overline{\lambda}^{k+1}-\overline{\lambda}^* \end{bmatrix}\right\|\right) \\
&\leq \; w^k\overline{C} \left\|\begin{bmatrix} x^k-x^* \\ \overline{\lambda}^k-\overline{\lambda}^* \end{bmatrix}\right\|_2 \\
&\qquad + (1-w^k)\left(\left\|\begin{bmatrix} x^k+\widehat{d}^k-x^\rho+x^\rho-x^* \\ \widehat{\lambda}^{k+1}-\lambda^\rho+\lambda^\rho-\overline{\lambda}^* \end{bmatrix}\right\| + \left\|\begin{bmatrix} x^k+\overline{d}^k-x^* \\ \overline{\lambda}^{k+1}-\overline{\lambda}^* \end{bmatrix}\right\|\right) \\
&\leq \; \overline{C} \left\|\begin{bmatrix} x^k-x^* \\ \overline{\lambda}^k-\overline{\lambda}^* \end{bmatrix}\right\|_2 \\
&\qquad + (1-w^k)\left(\left\|\begin{bmatrix} x^k+\widehat{d}^k-x^\rho \\ \widehat{\lambda}^{k+1}-\lambda^\rho \end{bmatrix}\right\| + \left\|\begin{bmatrix} x^\rho-x^* \\ \lambda^\rho-\overline{\lambda}^* \end{bmatrix}\right\|\right) \\
&\leq \; \overline{C} \left\|\begin{bmatrix} x^k-x^* \\ \overline{\lambda}^k-\overline{\lambda}^* \end{bmatrix}\right\|_2 + \widehat{C} \left\|\begin{bmatrix} x^k-x^\rho \\ \widehat{\lambda}^k-\lambda^\rho \end{bmatrix}\right\|_2 + O(\rho) \qquad (3.3.31)
\end{aligned}$$

Here, the second and fourth inequalities follow from Lemma 3.3.13(a), the third holds as we have simply augmented the latter two vector norms, and the last follows from

Lemmas 3.3.11 and 3.3.13(b). By applying Lemma 3.3.11, we also have that

$$\begin{aligned}
\left\|\begin{bmatrix} x^k - x^\rho \\ \widehat{\lambda}^k - \lambda^\rho \end{bmatrix}\right\|_2 &\leq \left\|\begin{bmatrix} x^k - x^* \\ \widehat{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|_2 + 2\left\|\begin{bmatrix} x^k - x^* \\ \widehat{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|\left\|\begin{bmatrix} x^\rho - x^* \\ \lambda^\rho - \overline{\lambda}^* \end{bmatrix}\right\| + \left\|\begin{bmatrix} x^\rho - x^* \\ \lambda^\rho - \overline{\lambda}^* \end{bmatrix}\right\|_2 \\
&\leq \left\|\begin{bmatrix} x^k - x^* \\ \widehat{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|_2 + O(\rho) \\
&\leq \left\|\begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|_2 + 2\|\widehat{\lambda}^k - \overline{\lambda}^k\| \left\|\begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\| + \|\widehat{\lambda}^k - \overline{\lambda}^k\|_2 + O(\rho) \\
&\leq \left\|\begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|_2 + O(\|\widehat{\lambda}^k - \overline{\lambda}^k\|) + O(\rho). \qquad (3.3.32)
\end{aligned}$$

By (3.3.31) and (3.3.32), we obtain

$$\left\|\begin{bmatrix} x^{k+1} - x^* \\ \overline{\lambda}^{k+1} - \overline{\lambda}^* \end{bmatrix}\right\| \leq (\overline{C} + \widehat{C}) \left\|\begin{bmatrix} x^k - x^* \\ \overline{\lambda}^k - \overline{\lambda}^* \end{bmatrix}\right\|_2 + O(\|\widehat{\lambda}^k - \overline{\lambda}^k\|) + O(\rho). \qquad (3.3.33)$$

Letting $C := \overline{C} + \widehat{C}$, we have shown (3.3.30). □

Local convergence to a KKT point

We now consider the local convergence of `SQuID` in the neighborhood of a KKT point for (3.2.1) satisfying Assumption 3.3.10. Our first result shows that in the neighborhood of a solution point, subproblem (3.2.7) yields a linearly feasible search direction, the penalty parameter remains constant, and the multipliers are not modified outside of the QO solves.

Lemma 3.3.14. *Suppose Assumption 3.3.10 holds. Then, for all sufficiently large k with $\|(x^k, \overline{\lambda}^k) - (x^*, \overline{\lambda}^*)\|$ and $\|(x^k, \widehat{\lambda}^k) - (x^*, \widehat{\lambda}^*)\|$ each sufficiently small:*

(a) A solution of (3.2.7) *has $(\overline{r}^k, \overline{s}^k, \overline{t}^k) = 0$, yielding $\mathcal{E}^k = \mathcal{E}$ and $\mathcal{I}^k = \mathcal{I}$;*

(b) ρ^k is not decreased by (3.2.13), (3.2.18), *or* (3.2.19), *and the multipliers $\widehat{\lambda}^k$ are not modified by* (3.2.14).

Proof. The proof of part (a) is similar to the proof of Lemma 3.3.12(a). That is, under Assumption 3.3.10 (which means that Assumption 3.3.9 holds), a solution of (3.2.7) with $(x^k, \overline{\lambda}^k)$ sufficiently close to $(x^*, \overline{\lambda}^*)$ has $\overline{d}^k$ yielding the same sets of positive, zero, and negative-valued equality and violated, active, and strictly satisfied inequality constraints as x^*. In this case, $\mathcal{Z}^* = \mathcal{E}$ and $\mathcal{S}^* \cup \mathcal{A}^* = \mathcal{I}$, so $(\overline{r}^k, \overline{s}^k, \overline{t}^k) = 0$.

Now consider part (b). If x^k is feasible, then $v(x^k) = 0$ and (3.2.12) is violated. On the other hand, if x^k is infeasible, then we have $\Delta l(\overline{d}^k, x^k) = v(x^k)$ by part (a), which implies (3.2.12) is violated again. Overall, these conclusions imply that (3.2.13) and (3.2.14) are both not triggered. As for (3.2.18) and (3.2.19), every time either of these updates is triggered, ρ^k is at least reduced by a fraction of its current value. Therefore, if either of these updates is triggered an infinite number of times, then we would have $\rho^k \to 0$. However, under Assumption 3.3.10 we have $\rho^k \to \rho^* > 0$, so for all sufficiently large k, ρ^k is not decreased by either update. □

Our second result is similar to Lemma 3.3.12; again, recall [70].

Lemma 3.3.15. *Suppose Assumption 3.3.10 holds. Then, for all sufficiently large k with $\|(x^k, \overline{\lambda}^k) - (x^*, \overline{\lambda}^*)\|$ and $\|(x^k, \widehat{\lambda}^k) - (x^*, \widehat{\lambda}^*)\|$ each sufficiently small, there is a local solution for* (3.2.9) *such that $\widehat{d}^k$ yields the same sets of active and strictly satisfied inequality constraints as x^*. Moreover, $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ satisfies*

$$\begin{bmatrix} H(x^k, \rho^*, \widehat{\lambda}^k) & \nabla c_{\mathcal{E}\cup\mathcal{A}^*}(x^k) \\ \nabla c_{\mathcal{E}\cup\mathcal{A}^*}(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} \widehat{d}^k \\ \widehat{\lambda}^{k+1}_{\mathcal{E}\cup\mathcal{A}^*} \end{bmatrix} = - \begin{bmatrix} \rho^* \nabla f(x^k) \\ c_{\mathcal{E}\cup\mathcal{A}^*}(x^k) \end{bmatrix} \tag{3.3.34}$$

and

$$\widehat{\lambda}^{k+1}_{\mathcal{A}^*} > 0 \quad \text{and} \quad \widehat{\lambda}^{k+1}_{\mathcal{S}^*} = 0. \tag{3.3.35}$$

Proof. By Lemma 3.3.14, we have $\mathcal{E}^k = \mathcal{E}$ and $\mathcal{I}^k = \mathcal{I}$ under the conditions of the lemma. Thus, with $\widehat{d}^k = 0$, the optimality conditions (3.3.2) reduce to (3.2.4), so (3.3.2) is solved at $(x^*, \widehat{\lambda}^*)$ by $(d, \lambda) = (0, \widehat{\lambda}^*)$. By (3.3.2d), $\widehat{\lambda}^*_{\mathcal{S}^*} = 0$. Hence, by (3.3.2a) and the definition of $\mathcal{A}^*$, the linear system (3.3.34) is satisfied at $(x^*, \widehat{\lambda}^*)$ by $(d, \lambda_{\mathcal{E}\cup\mathcal{A}^*}) = (0, \widehat{\lambda}^*_{\mathcal{E}\cup\mathcal{A}^*})$.

Under Assumption 3.3.9(a) and Assumption 3.3.10(c), the matrix in (3.3.34) is nonsingular at $(x^*, \widehat{\lambda}^*)$, and hence the solution of (3.3.34) varies continuously in a neighborhood of $(x^*, \widehat{\lambda}^*)$. In addition, under Assumption 3.3.10(c), $H(x^k, \rho^*, \widehat{\lambda}^k)$ in (3.3.34) is positive definite on the null space of $\nabla c_{\mathcal{E}\cup\mathcal{A}^*}(x^k)^T$ in a neighborhood of $(x^*, \widehat{\lambda}^*)$.

It follows from the conclusions in the previous paragraph that for all $(x^k, \widehat{\lambda}^k)$ sufficiently close to $(x^*, \widehat{\lambda}^*)$, the solution $(\widehat{d}^k, \widehat{\lambda}^{k+1}_{\mathcal{E}\cup\mathcal{A}^*})$ to (3.3.34) is sufficiently close to $(0, \widehat{\lambda}^*_{\mathcal{E}\cup\mathcal{A}^*})$ such that it satisfies

$$\widehat{\lambda}^{k+1}_{\mathcal{A}^*} > 0 \text{ and } c_{\mathcal{S}^*}(x^k) + \nabla c_{\mathcal{S}^*}(x^k)^T \widehat{d}^k < 0.$$

By construction, such a solution also satisfies (3.3.2) together with $\widehat{\lambda}^{k+1}_{\mathcal{S}^*} = 0$. Therefore, $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ is a KKT point of subproblem (3.2.9), and, as revealed above, it identifies the same sets of active and strictly satisfied inequality constraints as x^*. □

We are now prepared to prove our main theorem concerning the local convergence of `SQuID` in the neighborhood of KKT points for (3.2.1).

Theorem 3.3.12. *Suppose Assumption 3.3.10 holds. Then, for all large k with $\|(x^k, \overline{\lambda}^k) - (x^*, \overline{\lambda}^*)\|$ and $\|(x^k, \widehat{\lambda}^k) - (x^*, \widehat{\lambda}^*)\|$ each sufficiently small, $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ is obtained via (3.3.34), $d^k \leftarrow \widehat{d}^k$, and*

$$\left\| \begin{bmatrix} x^{k+1} - x^* \\ \widehat{\lambda}^{k+1} - \widehat{\lambda}^* \end{bmatrix} \right\| \leq C \left\| \begin{bmatrix} x^k - x^* \\ \widehat{\lambda}^k - \widehat{\lambda}^* \end{bmatrix} \right\|_2 \tag{3.3.36}$$

for some constant $C > 0$ independent of k.

Proof. By Lemma 3.3.15, under the conditions of the theorem, $(\widehat{d}^k, \widehat{\lambda}^{k+1})$ generated by subproblem (3.2.9) can be obtained via (3.3.34) and (3.3.35). This implies that d^k is a linearly feasible direction, so $w^k \leftarrow 0$ and $d^k \leftarrow \widehat{d}^k$. Therefore, since $H(x^k, \rho^*, \widehat{\lambda}^k) = G(x^k, \rho^*, \widehat{\lambda}^k_{\mathcal{E}\cup\mathcal{A}^*})$ in such cases, (3.3.34) (with $\widehat{d}$ interchanged with d^k) constitutes a Newton iteration applied to the nonlinear system $F(x, \rho^*, \lambda_{\mathcal{E}\cup\mathcal{A}^*}) = 0$ at $(x^k, \rho^*, \widehat{\lambda}^k)$. We can now apply standard Newton analysis. Under Assumption 3.3.8 we have that F is continuously differentiable and F' is Lipschitz continuous in a neighborhood of $(x^*, \rho^*, \widehat{\lambda}^*_{\mathcal{E}\cup\mathcal{A}^*})$. Moreover, under Assumption 3.3.9(a) and Assumption 3.3.10(c), the matrix F' is nonsingular at $(x^*, \rho^*, \widehat{\lambda}^*_{\mathcal{E}\cup\mathcal{A}^*})$, so its inverse exists and is bounded in norm in a neighborhood of $(x^*, \rho^*, \widehat{\lambda}^*_{\mathcal{E}\cup\mathcal{A}^*})$. By [28, Theorem 5.2.1], if $(x^k, \widehat{\lambda}^k)$ is sufficiently close to $(x^*, \widehat{\lambda}^*)$, then we have that (3.3.36) holds true. □

3.4 Numerical Experiments

In this section, we summarize the performance of `SQuID` as it was employed to solve collections of feasible and infeasible problem instances. Our code is a prototype Matlab implementation of Algorithm 1.

Mention of a few specifications of our implementation are appropriate before we present our numerical results. First, in order to avoid numerical issues caused by poor scaling of the problem functions, each function was scaled so that the ℓ_∞-norm of its gradient at the initial point was no larger than a given constant $g_{\max} > 0$. Moreover, our termination conditions are defined to take into account the magnitudes of the quantities involved in the computation of the optimality and feasibility errors. Specifically, we terminate and declare that an optimal solution has been found if

$$\mathcal{R}_{opt}(x^k, \rho^k, \widehat{\lambda}^{k+1}) \leq \gamma \max\{\chi_{opt,k}, 1\} \quad \text{and} \quad v_{inf}(x^k) \leq \gamma \max\{v_{inf}(x_0), 1\} \tag{3.4.1}$$

where $\gamma > 0$ is a given constant,

$$\begin{aligned} \chi_{opt,k} &:= \max\{\rho^k, \|\nabla f(x^k)\|_\infty, \|\nabla c_{\mathcal{E}}(x^k)\|_\infty, \|\nabla c_{\mathcal{I}}(x^k)\|_\infty, \|\widehat{\lambda}^{k+1}_{\mathcal{E}}\|_\infty, \|\widehat{\lambda}^{k+1}_{\mathcal{I}}\|_\infty\}, \\ \text{and} \quad v_{inf}(x^k) &:= \max\{\|c_{\mathcal{E}}(x^k)\|_\infty, \|\max\{c_{\mathcal{I}}(x^k), 0\}\|_\infty\}. \end{aligned}$$

We terminate and declare that an infeasible stationary point has been found if

$$\mathcal{R}_{inf}(x^k, \overline{\lambda}^{k+1}) \leq \gamma \max\{\chi_{inf,k}, 1\}, \; v_{inf}(x^k) > \gamma \max\{v_{inf}(x_0), 1\}, \; \text{and} \; \rho^k \leq \overline{\rho} \tag{3.4.2}$$

where $\overline{\rho} > 0$ is a given constant and

$$\chi_{inf,k} := \max\{\|\nabla c_{\mathcal{E}}(x^k)\|_\infty, \|\nabla c_{\mathcal{I}}(x^k)\|_\infty, \|\overline{\lambda}_{\mathcal{E}}^{k+1}\|_\infty, \|\overline{\lambda}_{\mathcal{I}}^{k+1}\|_\infty\}.$$

Despite the fact that Theorem 3.3.4 implies that we do not necessarily need $\rho^k \to 0$ when converging to an infeasible stationary point, we only terminate and declare infeasibility when ρ^k is sufficiently small, as specified in (3.4.2). This may lead to extra iterations being performed before infeasibility is declared, but aids the algorithm in avoiding declarations of infeasibility when applied to problem instances that are actually feasible. Since ρ^k is decreased rapidly in the neighborhood of an infeasible stationary point due to (3.2.12), the additional cost is worthwhile. We also take into account the scaling of the problem functions when considering whether a given point is sufficiently feasible so that subproblem (3.2.7) may be skipped. Specifically, if $v_{inf}(x^k) \leq \overline{\gamma} \max\{v_{inf}(x_0), 1\}$ for some $\overline{\gamma} > 0$, then we save computational expense by approximating the solution of subproblem (3.2.7) with $\overline{d}^k \leftarrow 0$ and $\overline{\lambda}^{k+1} \leftarrow \overline{\lambda}^k$.

Our implementation requires that subproblems (3.2.7) and (3.2.9) are convex, so we modify $H(x^k, 0, \overline{\lambda}^k)$ and $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$, if necessary, to make them positive definite. We do this by iteratively adding multiples of the identity matrix until the smallest computed eigenvalue is sufficiently positive. Specifically, if one of these matrices needs to be modified at iteration k, then with some $\xi > 0$ and an initial increment μ^k, we add $\mu^k I, \xi\mu^k I, \xi_2\mu^k I, \ldots$ until the smallest eigenvalue of the matrix is larger than a positive parameter $\mu_{\min}$. We then set $\mu^{k+1} \leftarrow \max\{\mu_{\min}, \psi\mu^k\}$ for some $\psi \in (0,1)$ to help save the computational expense of computing eigenvalues and modifying the matrix in the following iteration. If a matrix does not need to be modified during iteration k, then we reset $\mu^{k+1} \leftarrow \mu_{\min}$ for the following iteration. (We maintain different increments, μ_0^k and μ_ρ^k, for $H(x^k, 0, \overline{\lambda}^k)$ and $H(x^k, \widehat{\rho}^k, \widehat{\lambda}^k)$, respectively.) Of course, these modifications may slow the local convergence rate of the algorithm in the neighborhood of optimal solutions or infeasible stationary points that may fail to satisfy a strict second-order sufficiency condition, but they allow a prototype implementation such as ours to be well-defined when applied to nonconvex problems.

For computing the weight w^k required in (3.2.11) for iteration k, we initialize $w^k \leftarrow 0$ and check if (3.2.10) holds for $d^k \leftarrow \widehat{d}^k$. If it does, then the algorithm continues with these values for the weight and step, and otherwise we apply a bisection method to attempt to find the smallest root w^k of

$$\Psi(w) = \Delta l(w\overline{d}^k + (1-w)\widehat{d}^k; x^k) - \beta \Delta l(\overline{d}^k; x^k).$$

Since when $\overline{d}^k \neq 0$ we have $\Psi(1) > 0$ and $\Psi(0) < 0$, the bisection method is well-defined and their exists $w^k \in (0,1)$ such that $\Psi(w^k) = 0$. (Note that if $\overline{d}^k = 0$, then both $\overline{d}^k$ and $\widehat{d}^k$ will be linearly feasible, and so (3.2.10) is satisfied with $w^k \leftarrow 0$.) We terminate the bisection method when the width of the current interval is less than 10_{-8}. This and our choice of $\omega \leftarrow (1 - 10_{-18})$ ensure that we always compute $w^k < \omega$, effectively making this threshold value inconsequential for our numerical experiments.

As final notes on the particulars of our implementation, we remark that (3.2.7) and (3.2.9) are solved using Matlab's built-in `quadprog` routine. Also, the parameter values used are those provided in Table 3.1.

Table 3.1: Input parameters for a prototype Matlab implemenation of Algorithm 1.

Parameter	ρ_0	β	θ	κ_ρ	κ_λ	ϵ	δ	η
Value	10_{-1}	10_{-2}	10_{-1}	10	10	10_{-2}	$5 \times 10_{-1}$	10_{-8}
Parameter	$g_{\max}$	γ	$\overline{\rho}$	$\overline{\gamma}$	ξ	μ_0	ψ	$\mu_{\min}$
Value	10_2	10_{-6}	10_{-8}	10_{-8}	2	10_{-4}	10_{-1}	10_{-4}

We tested our implementation on 123 of the Hock-Schittkowski problems [49] available as AMPL models [38].[1] (Problems `hs068` and `hs069` were excluded from the original set of 125 problems as the required external function was not compiled.) The original versions of all of these problems are feasible, but we created a corresponding set of infeasible problems by adding the incompatible constraints $x_1 \leq 0$ and $x_1 \geq 1$, where x_1 is the first variable in the problem statement.

[1] `http://orfe.princeton.edu/~rvdb/ampl/nlmodels/cute/`

Termination results for our implementation applied to these problems are shown in Tables 3.2 and 3.3, which contain statistics for the feasible and infeasible problems, respectively. In Table 3.2, the "Succeed" column reveals the number and percentage of problems for which a point satisfying (3.4.1) was obtained, and the "Infeasible" column reveals those statistics for problems for which a point satisfying (3.4.2) was obtained. Similarly, the "Succeed" column in Table 3.3 reveals the number and percentage of problems for which a point satisfying (3.4.2) was obtained, and the "Feasible" column reveals those statistics for problems for which a point satisfying (3.4.1) was obtained. In both tables, a termination result in the latter of these two columns represents a situation where the algorithm failed to solve the problem correctly. Any time the algorithm fails to terminate within 10^3 iterations, the algorithm is deemed to "Fail". (Problem `hs112x` was excluded in the set of feasible problems due to a function evaluation error that occurred during the run.)

Table 3.2: Performance statistics of `SQuID` on feasible problems.

Problem type	Succeed	Fail	Infeasible	Total
Feasible	110 (90.16%)	11 (9.02%)	1 (0.82%)	122

Table 3.3: Performance statistics of `SQuID` on infeasible problems.

Problem type	Succeed	Fail	Feasible	Total
Infeasible	111 (90.24%)	12 (9.76%)	0 (0.0%)	123

From Tables 3.2 and 3.3, one can see that our code consistently attained a success rate of at least 90%, which is strong for a prototype implementation. In fact, for most of the failures and for the feasible problem that was reported to be infeasible, we found the problems to be very nonconvex. This led to excessive modifications of the Hessian matrices, and in many cases search directions that were poorly scaled. The results may be improved with a more sophisticated Hessian modification strategy and/or the incorporation of second-order correction steps.

We conclude our discussion of this set of numerical experiments by illustrating the local convergence behavior of `SQuID` on these sets of test problems. For those instances that are successfully solved within the iteration limit, we store the logarithms of $\mathcal{R}_{opt}$ and $\mathcal{R}_{inf}$ for the last 10 iterations for the feasible and infeasible problem instances and plot them in Figures 3.1 and 3.2, respectively. In the plots, T represents the last iteration for each run. (If a given problem is solved in fewer than 10 iterations, then its corresponding plot begins in the middle of the graph.) In Figures 3.1 and 3.2, one can see that most of the curves turn significantly downward on the right-hand side of the graph. The curves with a slope less than -1 over the last iterations indicate local superlinear convergence, and the curves with slope less than -2 indicate quadratic convergence. One finds that many of the curves possess slopes of this type, providing empirical evidence for the convergence results in §3.3.

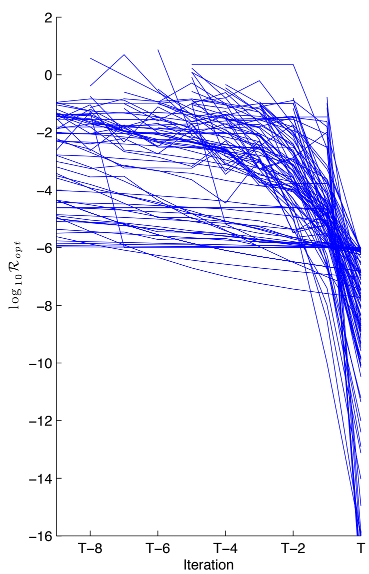


Figure 3.1: $\log_{10}\mathcal{R}_{opt}$ for the last 10 iterations of `SQuID` applied to feasible instances.

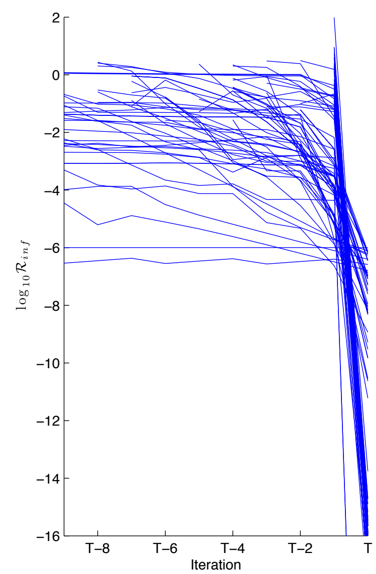


Figure 3.2: $\log_{10}\mathcal{R}_{inf}$ for the last 10 iterations of `SQuID` applied to infeasible instances.

We also compare the performance of `SQuID` on the 9 small-scale examples used in Chapter 2, where the first 8 examples are infeasible and the last one is infeasible. The numbers of iterations and function evaluations of `SQuID` are added to Table 2.2 to form the following Table 3.4, where a number with an asteroid superscript means the solver fails to solve the problem. This failure includes three cases: the solver runs out of iterations (Problem 2 and 7 for `Direct`, and Problem 7 for `Active`), an infeasible problem is reported feasible (Problem 5 and 6 for `Direct`), or an feasible problem is reported infeasible (Problem 9 for `Filter`). From the results shown in Table 3.4, one can see `SQuID` exhibits more stable performance than other solvers. It successfully solves all the examples, but need fewer iterations than `Ipopt`.

Table 3.4: Performance measures for solving the NLO problem.

Problem	`SQuID`		`Ipopt`		`Direct`		`Active`		`Filter`	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	29	30	48	281	38	135	22	235	16	16
2	23	31	109	170	*10000	*10000	23	167	12	12
3	23	29	788	3129	12	83	9	202	10	10
4	24	33	46	105	25	61	10	201	11	11
5	48	96	72	266	*1060	*3401	18	45	26	26
6	30	70	63	141	*76	*264	16	37	27	27
7	22	68	87	152	*10000	*43652	*10000	*20091	30	30
8	44	77	104	206	33	97	41	560	28	28
9	23	24	60	135	30	33	16	31	*13	*2

We close this section with a comparison between `SQuID` and the algorithm proposed in [13] when applied to solve the infeasible problems presented in [13]. As previously mentioned in §3.1.1, the algorithm in [13] represents an immediate predecessor of `SQuID` . That algorithm also possesses superlinear convergence guarantees, but, unlike `SQuID` , suffers from the disadvantage that more than 2 QO subproblem solves may be required in each iteration. After modifying the input parameters in our implementation of `SQuID` so that they match those used in [13]—e.g., in [13], the initial penalty parameter was set to 1—we obtained the results presented in Table 3.5. (Here, the "Iter." columns indicate the numbers of (nonlinear) iterations performed and the "QOs" columns indicate the number of QO subproblems solved prior to termination.) It is clear in these results that both algorithms detect infeasibility (or locate an optimal solution in the case of problem "batch") in few iterations, but `SQuID` typically requires fewer QO solves. (The only exception is the problem "robot". This problem is nonconvex and the performance of both algorithms varies depending on the input parameters that affect the modifications of the Hessian approximations to make them positive definite.) These results provide evidence for our claim that `SQuID` yields consistent improvement over the algorithm in [13]. That is, `SQuID` possesses similar theoretical convergence guarantees, but yields better practical performance by limiting the number of QO subproblem solves per iteration.

Table 3.5: Performance measures for test problems in [13]

	unique		robot		isolated		batch		batch1		nactive	
Alg.	Iter.	QOs	Iter.	QOs	Iter.	QOs	Iter.	QOs	Iter.	QOs	Iter.	QOs
`SQuID`	9	19	27	55	9	19	10	22	15	31	7	15
Ref 5	9	24	13	34	7	20	11	28	15	40	6	17

Chapter 4

Matrix-Free Solvers for Exact Penalty Subproblems

4.1 Introduction

irwa.introduction

The prototypical convex composite optimization problem is

$$\min_{x \in X} \ f(x) + \operatorname{dist}\left(F(x) \mid C\right), \tag{4.1.1}$$

where the sets $X \subset \mathbb{R}^n$ and $C \subset \mathbb{R}^m$ are non-empty, closed, and convex, the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $F : \mathbb{R}^n \to \mathbb{R}^m$ are smooth, and the distance function is defined as

$$\operatorname{dist}\left(y \mid C\right) := \inf_{z \in C} \ \|y - z\|,$$

with $\|\cdot\|$ a given norm on $\mathbb{R}^m$ [6, 33, 69]. The objective in problem (4.1.1) is an exact penalty function for the optimization problem

$$\min_{x \in X} \ f(x) \text{ subject to } F(x) \in C,$$

where the penalty parameter has been absorbed into the distance function. Problem (4.1.1) is also useful in the study of feasibility problems where one takes $f \equiv 0$.

Problems of the form (4.1.1) and algorithms for solving them have received a great deal of study over the last 30 years [1, 35, 72]. The typical approach for solving such problems is to apply a Gauss-Newton strategy to either define a direction-finding subproblem paired with a line search, or a trust-region subproblem to define a step to a new point [6, 69]. This chapter concerns the design, analysis, and implementation of methods for approximately solving the subproblems in either type of approach in large-scale settings. These subproblems take the form

$$\min_{x \in X} \; g^T x + \tfrac{1}{2} x^T H x + \operatorname{dist}\left(Ax + b \mid C\right), \tag{4.1.2}$$

where $g \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ is symmetric, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $X \subset \mathbb{R}^n$ and $C \subset \mathbb{R}^m$ may be modified versions of the corresponding sets in (4.1.1). In particular, the set X may now include the addition of a trust-region constraint. In practice, the matrix H is an approximation to the Hessian of the Lagrangian for the problem (4.1.1) [10, 33, 69], and so may be indefinite depending on how it is formed. However, in this chapter, we assume that it is positive semi-definite so that subproblem (4.1.2) is convex.

To solve large-scale instances of (4.1.2), we develop two solution methods based on linear least-squares subproblems. These solution methods are matrix-free in the sense that the least-squares subproblems can be solved in a matrix-free manner. The first approach is a novel iterative re-weighting strategy [2, 63, 65, 76, 83], while the second is based on ADAL technology [5, 30, 77] adapted to this setting. We prove that both algorithms are globally convergent under loose assumptions, and that each requires at most $O(1/\varepsilon^2)$ iterations to reach ε-optimality of the objective of (4.1.2). We conclude with numerical experiments that compare these two approaches.

As a first refinement, we suppose that C has the product space structure

$$C := C_1 \times \cdots \times C_l, \tag{4.1.3}$$

where, for each $i \in \mathcal{I} := \{1, 2, \ldots, l\}$, the set $C_i \subset \mathbb{R}^{m_i}$ is convex and $\sum_{i \in \mathcal{I}} m_i = m$. Conformally decomposing A and b, we write

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_l \end{bmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_l \end{pmatrix},$$

where, for each $i \in \mathcal{I}$, we have $A_i \in \mathbb{R}^{m_i \times n}$ and $b_i \in \mathbb{R}^{m_i}$. On the product space $\mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_l}$, we define a norm adapted to this structure as

$$\left\| (y_1^T, y_2^T, \ldots, y_l^T)^T \right\| := \sum_{i \in \mathcal{I}} \|y_i\|_2 \,. \tag{4.1.4}$$

It is easily verified that the corresponding dual norm is

$$\|y\|_* = \sup_{i \in \mathcal{I}} \|y_i\|_2 \,.$$

With this notation, we may write

$$\operatorname{dist}(y \mid C) = \sum_{i \in \mathcal{I}} \operatorname{dist}_2(y_i \mid C_i), \tag{4.1.5}$$

where, for any set S, we define the distance function $\operatorname{dist}_2(y \mid S) := \inf_{z \in S} \|y - z\|_2$. Hence, with $\varphi(x) := g^T x + \frac{1}{2} x^T H x$, subproblem (4.1.2) takes the form

$$\min_{x \in X} J_0(x), \quad \text{where} \quad J_0(x) := \varphi(x) + \sum_{i \in \mathcal{I}} \operatorname{dist}_2(A_i x + b_i \mid C_i) \,. \tag{4.1.6}$$

Throughout our algorithm development and analysis, it is important to keep in mind that $\|y\| \neq \|y\|_2$ since we make heavy use of *both* of these norms.

Example 4.1.1 (Intersections of Convex Sets)**.** *In many applications, the affine constraint has the representation* $\hat{A}x + \hat{b} \in \hat{C} := \bigcap_{i \in \mathcal{I}} C_i$*, where* $C_i \subset \mathbb{R}^{m_i}$ *is non-empty, closed, and convex for each* $i \in \mathcal{I}$*. Problems of this type are easily modeled in our framework by setting*

$A_i := \hat{A}$ and $b_i := \hat{b}$ for each $i \in \mathcal{I}$, and $C := C_1 \times \cdots \times C_l$.

4.1.1 Notation

Much of the notation that we use is standard and based on that employed in [72]. For convenience, we review some of this notation here. The set $\mathbb{R}^n$ is the real n-dimensional Euclidean space with $\mathbb{R}^n_+$ being the positive orthant in $\mathbb{R}^n$ and $\mathbb{R}^n_{++}$ the interior of $\mathbb{R}^n_+$. The set of real $m \times n$ matrices will be denoted as $\mathbb{R}^{m\times n}$. The Euclidean norm on $\mathbb{R}^n$ is denoted $\|\cdot\|_2$, and its closed unit ball is $\mathbb{B}_2 := \{x \mid \|x\|_2 \leq 1\}$. The closed unit ball of the norm defined in (4.1.4) will be denoted by $\mathbb{B}$. Vectors in $\mathbb{R}^n$ will be considered as column vectors and so we can write the standard inner product on $\mathbb{R}^n$ as $\langle u,\, v\rangle := u^T v$ for all $\{u, v\} \subset \mathbb{R}^n$. The set $\mathbb{N}$ is the set of natural numbers $\{1, 2, \dots\}$. Given $\{u, v\} \subset \mathbb{R}^n$, the line segment connecting them is denoted by $[u, v]$. Given a set $X \subset \mathbb{R}^n$, we define the convex indicator for X by

$$\delta\left(x \mid X\right) := \begin{cases} 0 & \text{if } x \in X, \\ +\infty & \text{if } x \notin X, \end{cases}$$

and its support function by

$$\delta^*\left(y \mid X\right) := \sup_{x \in X} \langle y,\, x\rangle\,.$$

A function $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R}^{\cup}\{+\infty\}$ is said to be convex if its epigraph,

$$\mathrm{epi}(f) := \{(x, \mu) \mid f(x) \leq \mu\}\,,$$

is a convex set. The function f is said to be closed (or lower semi-continuous) if epi(f) is closed, and f is said to be proper if $f(x) > -\infty$ for all $x \in \mathbb{R}^n$ and $\mathrm{dom}\,(f) := \{x \mid f(x) < \infty\} \neq \emptyset$. If f is convex, then the subdifferential of f at $\bar{x}$ is given by

$$\partial f(\bar{x}) := \{z \mid f(\bar{x}) + \langle z,\, x - \bar{x}\rangle \leq f(x)\ \forall\, x \in \mathbb{R}^n\}\,.$$

Given a closed convex $X \subset \mathbb{R}^n$, the normal cone to X at a point $\bar{x} \in X$ is given by

$$N(\bar{x} \,|\, X) := \{z \mid \langle z,\, x - \bar{x}\rangle \le 0 \;\forall\, x \in X\}.$$

It is well known that $N(\bar{x} \,|\, X) = \partial\delta(\bar{x} \mid X)$; e.g., see [72]. Given a set $S \subset \mathbb{R}^m$ and a matrix $M \in \mathbb{R}^{m\times n}$, the inverse image of S under M is given by

$$M^{-1}S := \{x \mid Mx \in S\}.$$

Since the set C in (4.1.3) is non-empty, closed, and convex, the distance function $\text{dist}(y \mid C)$ is convex. Using the techniques of [72], it is easily shown that the subdifferential of the distance function (4.1.5) is

$$\partial\text{dist}(p \mid C) = \partial\text{dist}_2(p_1 \mid C_1) \times \cdots \times \partial\text{dist}_2(p_l \mid C_l), \tag{4.1.7}$$

where, for each $i \in \mathcal{I}$, we have

$$\partial\text{dist}_2(p_i \mid C_i) = \begin{cases} \frac{(I-P_{C_i})p_i}{\left\|(I-P_{C_i})p_i\right\|_2} & \text{if } i \notin \mathcal{A}(p), \\ \mathbb{B}_2 \cap N(p_i \,|\, C_i) & \text{if } i \in \mathcal{A}(p). \end{cases} \tag{4.1.8}$$

Here, we have defined

$$\mathcal{A}(p) := \{i \in \mathcal{I} \mid \text{dist}_2(p_i \mid C_i) = 0\} \quad \forall\, p \in \mathbb{R}^m,$$

and let $P_C(p)$ denote the projection of p onto the set C (see Theorem 4.2.1).

Since we will be working on the product space $\mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_l}$, we will need notation for the components of the vectors in this space. Given a vector $w \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_l}$, we denote the components in $\mathbb{R}^{m_i}$ by w_i and the jth component of w_i by w_{ij} for $j = 1, \ldots, m_i$ and $i \in \mathcal{I}$ so that $w = (w_1^T, \ldots, w_l^T)^T$. Correspondingly, given vectors $w_i \in \mathbb{R}^{m_i}$ for $i \in \mathcal{I}$, we denote by $w \in \mathbb{R}^m$ the vector $w = (w_1^T, \ldots, w_l^T)^T$.

4.2 An Iterative Re-Weighting Algorithm

We now describe an iterative algorithm for minimizing the function J_0 in (4.1.6), where in each iteration one solves a subproblem whose objective is the sum of φ and a weighted linear least-squares term. An advantage of this approach is that the subproblems can be solved using matrix-free methods, e.g., the conjugate gradient (CG), projected gradient, and Lanczos [46] methods. The objectives of the subproblems are localized approximations to J_0 based on projections. In this manner, we will make use of the following theorem.

Theorem 4.2.1. *[85] Let $C \subset \mathbb{R}^m$ be non-empty, closed, and convex. Then, to every $y \in \mathbb{R}^m$, there is a unique $\bar{y} \in C$ such that*

$$\|y - \bar{y}\|_2 = \operatorname{dist}_2(y \mid C).$$

We call $\bar{y} = P_C(y)$ the projection of y onto C. Moreover, the following hold:

1. *$\bar{y} = P_C(y)$ if and only if $\bar{y} \in C$ and $(y - \bar{y}) \in N(\bar{y} \mid C)$ [73].*
2. *For all $\{y, z\} \subset \mathbb{R}^m$, the operator P_C yields*

$$\|P_C(y) - P_C(z)\|_2^2 + \|(I - P_C)y - (I - P_C)z\|_2^2 \leq \|y - z\|_2^2.$$

Since H is symmetric and positive semi-definite, there exists $A_0 \in \mathbb{R}^{m_0 \times n}$, where $m_0 := \operatorname{rank}(H)$, such that $H = A_0^T A_0$. We use this representation for H in order to simplify our mathematical presentation; this factorization is not required in order to implement our methods. Define $b_0 := 0 \in \mathbb{R}^n$, $C_0 := \{0\} \subset \mathbb{R}^n$, and $\mathcal{I}_0 := \{0\} \cup \mathcal{I} = \{0, 1, \ldots, l\}$. Using this notation, we define our local approximation to J_0 at a given point $\tilde{x}$ and with a given *relaxation vector* $\epsilon \in \mathbb{R}^l_{++}$ by

$$\hat{G}_{(\tilde{x},\epsilon)}(x) := g^T x + \tfrac{1}{2} \sum_{i \in \mathcal{I}_0} w_i(\tilde{x}, \epsilon) \|A_i x + b_i - P_{C_i}(A_i \tilde{x} + b_i)\|_2^2,$$

where, for any $x \in \mathbb{R}^n$, we define

$$w_0(x,\epsilon) := 1, \quad w_i(x,\epsilon) := \left(\text{dist}_2^2(A_i x + b_i \mid C_i) + \epsilon_i^2\right)^{-1/2} \quad \forall\, i \in \mathcal{I}, \tag{4.2.1}$$
$$\text{and} \quad W(x,\epsilon) := \text{diag}(w_0(x,\epsilon) I_{m_0}, \ldots, w_l(x,\epsilon) I_{m_l}).$$

Define

$$\tilde{A} := \begin{bmatrix} A_0 \\ A \end{bmatrix}. \tag{4.2.2}$$

We now state the algorithm.

<u>Iterative Re-Weighting Algorithm (IRWA)</u>

Step 0: (Initialization) Choose an initial point $x^0 \in X$, an initial relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$, and scaling parameters $\eta \in (0,1)$, $\gamma > 0$, and $M > 0$. Let $\sigma \geq 0$ and $\sigma' \geq 0$ be two scalars which serve as termination tolerances for the stepsize and relaxation parameter, respectively. Set $k := 0$.

Step 1: (Solve the re-weighted subproblem for x^{k+1})

Compute a solution x^{k+1} to the problem

$$\mathcal{G}(x^k, \epsilon^k) : \quad \min_{x \in X} \hat{G}_{(x^k,\epsilon^k)}(x). \tag{4.2.3}$$

Step 2: (Set the new relaxation vector ϵ^{k+1})

Set

$$q_i^k := A_i(x^{k+1} - x^k) \;\text{ and }\; r_i^k := (I - P_{C_i})(A_i x^k + b_i) \quad \forall\, i \in \mathcal{I}_0.$$

If

$$\left\| q_i^k \right\|_2 \leq M \left[\left\| r_i^k \right\|_2^2 + (\epsilon_i^k)^2 \right]^{\frac{1}{2}+\gamma} \quad \forall\, i \in \mathcal{I}, \tag{4.2.4}$$

then choose $\epsilon^{k+1} \in (0,\ \eta\epsilon^k]$; else, set $\epsilon^{k+1} := \epsilon^k$.

Step 3: (Check stopping criteria)

If $\left\| x^{k+1} - x^k \right\|_2 \leq \sigma$ and $\left\| \epsilon^k \right\|_2 \leq \sigma'$, then stop; else, set $k := k+1$ and go to Step 1.

Remark 4.2.2. In cases where $C_i = \{0\} \subset \mathbb{R}$ for all $i \in \mathcal{I}$ and $\phi \equiv 0$, this algorithm has a long history in the literature. Two early references are [2] and [76]. In such cases, the algorithm reduces to the classical algorithm for minimizing $\|Ax + b\|_1$ using iteratively re-weighted least-squares.

Remark 4.2.3. If there exists z_0 such that $A_0^T z_0 = g$, then, by setting $b_0 := z_0$, the linear term $g^T x$ can be eliminated in the definition of $\hat{G}$.

Remark 4.2.4. It is often advantageous to employ a stopping criteria based on a percent reduction in the duality gap rather than the stopping criteria given in Step 3 above [6, 12]. In such cases, one keeps track of both the primal objective values $J_0^k := J_0(x^k)$ and the dual objective values

$$\hat{J}_0^k := \tfrac{1}{2}(g + A^T\tilde{u}^k)^T H^{-1}(g + A^T\tilde{u}^k) - b^T\tilde{u}^k + \sum_{i\in\mathcal{I}} \delta^*\left(\tilde{u}_i^k \mid C_i\right),$$

where the vectors $\tilde{u}^k := W_k r^k$ are dual feasible (see (4.5.2) for a discussion of the dual problem). Given $\sigma \in (0, 1)$, Step 3 above can be replaced by

Step 3': (Check stopping criteria)

If $(J_0^1 + \hat{J}_0^k) \le \sigma(J_0^1 - J_0^k)$, then stop; else, set $k := k + 1$ and go to Step 1.

This is the stopping criteria employed in some of our numerical experiments. Nonetheless, for our analysis, we employ Step 3 as it is stated in the formal description of IRWA for those instances when dual values $\hat{J}_0^k$ are unavailable, such as when these computations are costly or subject to error.

4.2.1 Smooth Approximation to J_0

Our analysis of IRWA is based on a smooth approximation to J_0. Given $\epsilon \in \mathbb{R}^l_+$, define the ϵ-smoothing of J_0 by

$$J(x, \epsilon) := \varphi(x) + \sum_{i\in\mathcal{I}} \sqrt{\operatorname{dist}_2^2(A_i x + b_i \mid C_i) + \epsilon_i^2}\ . \tag{4.2.5}$$

Note that $J_0(x) \equiv J(x,0)$ and that $J(x,\epsilon)$ is jointly convex in (x,ϵ) since

$$J(x,\epsilon) = \varphi(x) + \sum_{i\in\mathcal{I}} \mathrm{dist}_2\left(\begin{bmatrix} A_i & 0 \\ 0 & e_i^T \end{bmatrix} \begin{pmatrix} x \\ \epsilon \end{pmatrix} + \begin{pmatrix} b_i \\ 0 \end{pmatrix} \mid C_i \times \{0\} \right),$$

where e_i is the ith unit coordinate vector. By [74, Corollary 10.11], (4.1.7), and (4.1.8),

$$\partial J_0(x) = \partial_x J(x,0) = \nabla\varphi(x) + A^T \partial \mathrm{dist}\left(\cdot \mid C\right)(Ax+b) = \tag{4.2.6}$$
$$\nabla\varphi(x) + \sum_{i\notin\mathcal{A}(Ax+b)} A_i^T \frac{(I-P_{C_i})(A_i x + b_i)}{\|(I-P_{C_i})(A_i x+b_i)\|_2} + \sum_{i\in\mathcal{A}(Ax+b)} A_i^T (\mathbb{B}_2 \cap N\left(A_i x + b_i \,|\, C_i\right)).$$

Given $\tilde{x} \in \mathbb{R}^n$ and $\tilde{\epsilon} \in \mathbb{R}^l_{++}$, we define a weighted approximation to $J(\cdot,\tilde{\epsilon})$ at $\tilde{x}$ by

$$G_{(\tilde{x},\tilde{\epsilon})}(x) := g^T x + \tfrac{1}{2}\sum_{i\in\mathcal{I}_0} w_i(\tilde{x},\tilde{\epsilon}) \mathrm{dist}_2^2\left(A_i x + b_i \mid C_i\right).$$

We have the following fundamental fact about solutions of $\mathcal{G}(\tilde{x},\tilde{\epsilon})$ defined by (4.2.3).

Lemma 4.2.1. *Let $\tilde{x} \in X$, $\tilde{\epsilon} \in \mathbb{R}^l_{++}$, $\hat{\epsilon} \in (0,\tilde{\epsilon}]$, and $\hat{x} \in \operatorname{argmin}_{x\in X} \hat{G}_{(\tilde{x},\tilde{\epsilon})}(x)$. Set $\tilde{w}_i := w_i(\tilde{x},\tilde{\epsilon})$ and $q_i := A_i(\hat{x}-\tilde{x})$ for $i \in \mathcal{I}_0$, $\tilde{W} := W(\tilde{x},\tilde{\epsilon})$, and $q := (q_0^T,\dots,q_l^T)^T$. Then,*

$$G_{(\tilde{x},\tilde{\epsilon})}(\hat{x}) - G_{(\tilde{x},\tilde{\epsilon})}(\tilde{x}) \le -\tfrac{1}{2} q^T \tilde{W} q \tag{4.2.7}$$

and

$$J(\tilde{x},\tilde{\epsilon}) - J(\hat{x},\hat{\epsilon}) \ge \tfrac{1}{2} q^T \tilde{W} q. \tag{4.2.8}$$

Proof. We first prove (4.2.7). Define $\hat{r}_i := (I-P_{C_i})(A_i\hat{x}+b_i)$ and $\tilde{r}_i := (I-P_{C_i})(A_i\tilde{x}+b_i)$ for $i \in \mathcal{I}_0$, and set $\hat{r} := (\hat{r}_0^T,\dots,\hat{r}_l^T)^T$ and $\tilde{r} := (\tilde{r}_0^T,\dots,\tilde{r}_l^T)^T$. Since $\hat{x} \in \operatorname{argmin}_{x\in X} \hat{G}_{(\tilde{x},\tilde{\epsilon})}(x)$, there exists $\hat{v} \in N\left(\hat{x} \,|\, X\right)$ such that

$$0 = g + \tilde{A}^T \tilde{W}(\tilde{A}\hat{x} + b - P_C(\tilde{A}\tilde{x}+b)) + \hat{v} = g + \tilde{A}^T\tilde{W}(q+\tilde{r}) + \hat{v},$$

or, equivalently,

$$-\hat{v} = g + \tilde{A}^T \tilde{W}(q + \tilde{r}). \tag{4.2.9}$$

Moreover, by the definition of the projection operator P_{C_i}, we know that

$$\|\hat{r}_i\|_2 = \|(I - P_{C_i})(A_i\hat{x} + b_i)\|_2 \leq \|A_i\hat{x} + b_i - P_{C_i}(A_i\tilde{x} + b_i)\|_2 = \|q_i + \tilde{r}_i\|_2$$

so that

$$\|\hat{r}_i\|_2^2 - \|q_i + \tilde{r}_i\|_2^2 \leq 0 \quad \forall\, i \in \mathcal{I}_0. \tag{4.2.10}$$

Therefore,

$$\begin{aligned}
& G_{(\tilde{x},\tilde{\epsilon})}(\hat{x}) - G_{(\tilde{x},\tilde{\epsilon})}(\tilde{x}) \\
= {} & g^T(\hat{x} - \tilde{x}) + \tfrac{1}{2}\textstyle\sum_{i\in\mathcal{I}_0} \tilde{w}_i[\|\hat{r}_i\|_2^2 - \|\tilde{r}_i\|_2^2] \\
= {} & g^T(\hat{x} - \tilde{x}) + \tfrac{1}{2}\textstyle\sum_{i\in\mathcal{I}_0} \tilde{w}_i[(\|\hat{r}_i\|_2^2 - \|q_i + \tilde{r}_i\|_2^2) + (\|q_i + \tilde{r}_i\|_2^2 - \|\tilde{r}_i\|_2^2)] \\
\leq {} & g^T(\hat{x} - \tilde{x}) + \tfrac{1}{2}\textstyle\sum_{i\in\mathcal{I}_0} \tilde{w}_i[\|q_i + \tilde{r}_i\|_2^2 - \|\tilde{r}_i\|_2^2] && \text{(by (4.2.10))} \\
= {} & g^T(\hat{x} - \tilde{x}) + \tfrac{1}{2}\textstyle\sum_{i\in\mathcal{I}_0} \tilde{w}_i[\|q_i\|_2^2 + 2\langle q_i,\, \tilde{r}_i\rangle] \\
= {} & g^T(\hat{x} - \tilde{x}) + \tfrac{1}{2}\textstyle\sum_{i\in\mathcal{I}_0} \tilde{w}_i[-\|q_i\|_2^2 + 2\langle q_i,\, q_i + \tilde{r}_i\rangle] \\
= {} & -\tfrac{1}{2}q^T\tilde{W}q + g^T(\hat{x} - \tilde{x}) + q^T\tilde{W}(q + \tilde{r}) \\
= {} & -\tfrac{1}{2}q^T\tilde{W}q + (\hat{x} - \tilde{x})^T(g + \tilde{A}^T\tilde{W}(q + \tilde{r})) \\
= {} & -\tfrac{1}{2}q^T\tilde{W}q + (\tilde{x} - \hat{x})^T\hat{v} && \text{(by (4.2.9))} \\
\leq {} & -\tfrac{1}{2}q^T\tilde{W}q,
\end{aligned}$$

where the final inequality follows since $\tilde{x} \in X$ and $\hat{v} \in N(\hat{x} \,|\, X)$.

We now prove (4.2.8). Since $\sqrt{t}$ is a concave function of t on $\mathbb{R}_+$, we have

$$\sqrt{\hat{t}} \leq \sqrt{\tilde{t}} + \frac{\hat{t} - \tilde{t}}{2\sqrt{\tilde{t}}} \quad \forall\, \{\hat{t}, \tilde{t}\} \subset \mathbb{R}_{++},$$

and so, for $i \in \mathcal{I}$, we have

$$\begin{aligned} &\sqrt{\operatorname{dist}_2^2(A_i\hat{x} + b_i \mid C_i) + \tilde{\epsilon}_i^2} \\ &\le \sqrt{\operatorname{dist}_2^2\left(A_i\tilde{x} + b_i \mid C_i\right) + \tilde{\epsilon}_i^2} + \frac{\operatorname{dist}_2^2\left(A_i\hat{x} + b_i \mid C_i\right) - \operatorname{dist}_2^2\left(A_i\tilde{x} + b_i \mid C_i\right)}{2\sqrt{\operatorname{dist}_2^2\left(A_i\tilde{x} + b_i \mid C_i\right) + \tilde{\epsilon}_i^2}}. \end{aligned} \tag{4.2.11}$$

Hence,

$$\begin{aligned} J(\hat{x}, \hat{\epsilon}) \le J(\hat{x}, \tilde{\epsilon}) &= \varphi(\hat{x}) + \sum_{i \in \mathcal{I}} \sqrt{\operatorname{dist}_2^2(A_i\hat{x} + b_i \mid C_i) + \tilde{\epsilon}_i^2} \\ &\le J(\tilde{x}, \tilde{\epsilon}) + (\varphi(\hat{x}) - \varphi(\tilde{x})) + \tfrac{1}{2} \sum_{i \in \mathcal{I}} \frac{\operatorname{dist}_2^2\left(A_i\hat{x} + b_i \mid C_i\right) - \operatorname{dist}_2^2\left(A_i\tilde{x} + b_i \mid C_i\right)}{\sqrt{\operatorname{dist}_2^2\left(A_i\tilde{x} + b_i \mid C_i\right) + \tilde{\epsilon}_i^2}} \\ &= J(\tilde{x}, \tilde{\epsilon}) + [G_{(\tilde{x},\tilde{\epsilon})}(\hat{x}) - G_{(\tilde{x},\tilde{\epsilon})}(\tilde{x})] \\ &\le J(\tilde{x}, \tilde{\epsilon}) - \tfrac{1}{2} q^T \tilde{W} q, \end{aligned}$$

where the first inequality follows from $\hat{\epsilon} \in (0, \tilde{\epsilon}]$, the second inequality follows from (4.2.11), and the third inequality follows from (4.2.7). □

4.2.2 Coercivity of J

Lemma 4.2.1 tells us that IRWA is a descent method for the function J. Consequently, both the existence of solutions to (4.1.6) as well as the existence of cluster points to IRWA can be guaranteed by understanding conditions under which the function J is coercive, or equivalently, conditions that guarantee the boundedness of the lower level sets of J over X. For this, we need to consider the asymptotic geometry of J and X.

Definition 4.2.5. [74, Definition 3.3] Given $Y \subset \mathbb{R}^m$, the horizon cone of Y is

$$Y^\infty := \left\{ z \;\middle|\; \exists\, t^k \downarrow 0,\ \{y^k\} \subset Y \text{ such that } t^k y^k \to z \right\}.$$

We have the basic facts about horizon cones given in the following proposition.

Proposition 4.2.6. *The following hold:*

1. The set $Y \subset \mathbb{R}^m$ *is bounded if and only if* $Y^\infty = \{0\}$.

2. Given $Y_i \subset \mathbb{R}^{m_i}$ *for* $i \in \mathcal{I}$, *we have* $(Y_1 \times \cdots \times Y_l)^\infty = Y_1^\infty \times \cdots \times Y_l^\infty$.

3. [74, Theorem 3.6] If $C \subset \mathbb{R}^m$ *is non-empty, closed, and convex, then*

$$C^\infty = \{z \mid C + z \subset C\}.$$

We now prove the following result about the lower level sets of J.

Theorem 4.2.7. *Let* $\alpha > 0$ *and* $\epsilon \in \mathbb{R}^l_+$ *be such that the set*

$$L(\alpha, \epsilon) := \{x \in X \mid J(x, \epsilon) \leq \alpha\}$$

is non-empty. Then,

$$L(\alpha, \epsilon)^\infty = \left\{\bar{x} \in X^\infty \;\middle|\; g^T\bar{x} \leq 0,\ H\bar{x} = 0,\ A\bar{x} \in C^\infty\right\}. \tag{4.2.12}$$

Moreover, $L(\alpha, \epsilon)$ *is compact for all* $(\alpha, \epsilon) \in \mathbb{R}^{l+1}_+$ *if and only if*

$$\left[\bar{x} \in X^\infty \cap \ker(H) \cap A^{-1}C^\infty \text{ satisfies } g^T\bar{x} \leq 0\right] \iff \bar{x} = 0. \tag{4.2.13}$$

Proof. Let $x \in L(\alpha, \epsilon)$ and let $\bar{x}$ be an element of the set on the right-hand side of (4.2.12). Then, by Proposition 4.2.6, for all $\lambda \geq 0$ we have $x + \lambda\bar{x} \in X$ and $\lambda A_i\bar{x} + C_i \subset C_i$ for all $i \in \mathcal{I}$, and so for each $i \in \mathcal{I}$ we have

$$\begin{aligned}
\operatorname{dist}(A_i(x + \lambda\bar{x}) + b_i \mid C_i) &\leq \operatorname{dist}(A_i(x + \lambda\bar{x}) + b_i \mid \lambda A_i\bar{x} + C_i) \\
&= \operatorname{dist}((A_ix + b_i) + \lambda A_i\bar{x} \mid \lambda A_i\bar{x} + C_i) \\
&= \operatorname{dist}(A_ix + b_i \mid C_i).
\end{aligned}$$

Therefore,

$$\begin{aligned} J(x+\lambda\bar{x},\epsilon) &= \varphi(x)+\lambda g^T\bar{x}+\sum_{i\in\mathcal{I}}\sqrt{\mathrm{dist}_2^2\left(A_i(x+\lambda\bar{x})+b_i\mid C_i\right)+\epsilon_i^2} \\ &\le \varphi(x)+\sum_{i\in\mathcal{I}}\sqrt{\mathrm{dist}_2^2\left(A_ix+b_i\mid C_i\right)+\epsilon_i^2}=J(x,\epsilon)\le\alpha. \end{aligned}$$

Consequently, $\bar{x}\in L(\alpha,\epsilon)^\infty$.

On the other hand, let $\bar{x}\in L(\alpha,\epsilon)^\infty$. We need to show that $\bar{x}$ is an element of the set on the right-hand side of (4.2.12). For this, we may as well assume that $\bar{x}\neq 0$. By the fact that $\bar{x}\in L(\alpha,\epsilon)^\infty$, there exists $t^k\downarrow 0$ and $\{x^k\}\subset X$ such that $J(x^k,\epsilon)\le\alpha$ and $t^kx^k\to\bar{x}$. Consequently, $\bar{x}\in X^\infty$. Moreover,

$$g^T(t^kx^k)=t^k(g^Tx^k)\le t^kJ(x^k,\epsilon)\le t^k\alpha\to 0$$

and so

$$\begin{aligned} 0\le \left\|A_0(t^kx^k)\right\|^2 &= (t^kx^k)^TH(t^kx^k)=(t^k)^2(x^k)^THx^k \\ &\le (t^k)^2 2(J(x^k,\epsilon)-g^Tx^k)\le (t^k)^2 2\alpha - t^k 2g^T(t^kx^k)\to 0. \end{aligned}$$

Therefore, $g^T\bar{x}\le 0$ and $H\bar{x}=0$. Now, define $z^k:=P_C(Ax^k+b)$ for $k\in\mathbb{N}$. Then, by Theorem 4.2.1(2), we have

$$\left\|z^k\right\|_2\le\left\|(I-P_C)(Ax^k+b)\right\|_2+\left\|Ax^k+b\right\|_2\le\alpha+\left\|Ax^k+b\right\|_2,$$

which, since $A(t^kx^k)+t^kb\to A\bar{x}$, implies that the sequence $\{t^kz^k\}$ is bounded. Hence, without loss of generality, we can assume that there is a vector $\bar{z}$ such that $t^kz^k\to\bar{z}$, where by the definition of z^k we have $\bar{z}\in C^\infty$. But,

$$\begin{aligned} 0\le\left\|A(t^kx^k)+t^kb-(t^kz^k)\right\|_2 &= t^k\mathrm{dist}_2\left(Ax^k+b\mid C\right) \\ &\le t^kJ(x^k,\epsilon)-t^kg^Tx^k\le t^k\alpha-g^T(t^kx^k)\to 0, \end{aligned}$$

while

$$\left\| A(t^k x^k) + b - (t^k z^k) \right\|_2 \to \|A\bar{x} - \bar{z}\|_2 .$$

Consequently, $\bar{x} \in X^\infty,\ g^T\bar{x} \leq 0,\ H\bar{x} = 0$, and $A\bar{x} \in C^\infty$, which together imply that $\bar{x}$ is in the set on the right-hand side of (4.2.12). □

Corollary 4.2.8. *Suppose that the sequence $\{(x^k, \epsilon^k)\}$ is generated by IRWA with initial point $x^0 \in X$ and relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$. Then, $\{x^k\}$ is bounded if* (4.2.13) *is satisfied, which follows if at least one of the following conditions holds:*

1. *X is compact.*

2. *H is positive definite.*

3. *C is compact and $X^\infty \cap \ker(H) \cap \ker(A) = \{0\}$.*

Remark 4.2.9. For future reference, observe that

$$\ker(H) \cap \ker(A) = \ker(\tilde{A}), \tag{4.2.14}$$

where $\tilde{A}$ is defined in (4.2.2).

4.2.3 Convergence of IRWA

We now return to our analysis of the convergence of IRWA by first proving the following lemma that discusses critical properties of the sequence of iterates computed in the algorithm.

Lemma 4.2.2. *Suppose that the sequence $\{(x^k, \epsilon^k)\}$ is generated by IRWA with initial point $x^0 \in X$ and relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$, and, for $k \in \mathbb{N}$, let q_i^k and r_i^k for $i \in \mathcal{I}_0$ be as defined in Step 2 of the algorithm with*

$$q^k := ((q_0^k)^T, \ldots, (q_l^k)^T)^T \quad \textit{and} \quad r^k := ((r_0^k)^T, \ldots, (r_l^k)^T)^T.$$

Moreover, for $k \in \mathbb{N}$, define

$$w_i^k := w_i(x^k, \epsilon^k) \text{ for } i \in \mathcal{I}_0 \quad \text{and} \quad W_k := W(x^k, \epsilon^k),$$

and set $S := \left\{k \,\middle|\, \epsilon^{k+1} \leq \eta\epsilon^k\right\}$. Then, the sequence $\{J(x^k, \epsilon^k)\}$ is monotonically decreasing. Moreover, either $\inf_{k\in\mathbb{N}} J(x^k, \epsilon^k) = -\infty$, in which case $\inf_{x\in X} J_0(x) = -\infty$, or the following hold:

1. $\sum_{k=0}^{\infty}(q^k)^T W_k q^k < \infty$.
2. $\epsilon^k \to 0$ *and* $H(x^{k+1} - x^k) \to 0$.
3. $W_k q^k \xrightarrow{S} 0$.
4. $w_i^k r_i^k = r_i^k / \sqrt{\|r_i^k\|_2^2 + \epsilon_i^k} \in \mathbb{B}_2 \cap N\left(P_{C_i}(A_i x^k + b_i) \,|\, C_i\right)$, $i \in \mathcal{I}$, $k \in \mathbb{N}$.
5. $-\tilde{A}^T W_k q^k \in (\nabla\varphi(x^k) + \sum_{i\in\mathcal{I}} A_i^T w_i^k r_i^k) + N\left(x^{k+1} \,|\, X\right)$, $k \in \mathbb{N}$.
6. *If* $\{\text{dist}\left(Ax^k + b \mid C\right)\}_{k\in S}$ *is bounded, then* $q^k \xrightarrow{S} 0$.

Proof. The fact that $\{J(x^k, \epsilon^k)\}$ is monotonically decreasing is an immediate consequence of the monotonicity of the sequence $\{\epsilon_k\}$, Lemma 4.2.1, and the fact that W_k is positive definite for all $k \in \mathbb{N}$. If $J(x^k, \epsilon^k) \to -\infty$, then $\inf_{x\in X} J_0(x) = -\infty$ since $J_0(x) = J(x, 0) \leq J(x, \epsilon)$ for all $x \in \mathbb{R}^n$ and $\epsilon \in \mathbb{R}^l_+$. All that remains is to show that Parts (1)–(6) hold when $\inf_{k\in\mathbb{N}} J(x^k, \epsilon^k) > -\infty$, in which case we may assume that the sequence $\{J(x^k, \epsilon^k)\}$ is bounded below. We define the lower bound $\tilde{J} := \inf_{k\in\mathbb{N}} J(x^k, \epsilon^k) = \lim_{k\in\mathbb{N}} J(x^k, \epsilon^k)$ for the remainder of the proof.

(1) By Lemma 4.2.1, for every positive integer $\bar{k}$ we have

$$\begin{aligned}
\tfrac{1}{2}\sum_{k=0}^{\bar{k}}(q^k)^T W_k q^k &\leq \sum_{k=0}^{\bar{k}}[J(x^k, \epsilon^k) - J(x^{k+1}, \epsilon^{k+1})] \\
&= J(x^0, \epsilon^0) - J(x^{\bar{k}+1}, \epsilon^{\bar{k}+1}) \\
&\leq J(x^0, \epsilon^0) - \tilde{J}.
\end{aligned}$$

Therefore, as desired, we have

$$\sum_{k=0}^{\infty}(q^k)^T W_k q^k \le 2(J(x^0,\epsilon^0)-\tilde{J}) < \infty.$$

(2) Since $\eta \in (0,1)$, if $\epsilon^k \nrightarrow 0$, then there exists an integer $\bar{k} \ge 0$ and a scalar $\bar{\epsilon} > 0$ such that $\epsilon^k = \bar{\epsilon}$ for all $k \ge \bar{k}$. Part (1) implies that $(q^k)^T W_k q^k$ is summable so that $(w_i^k \|q_i^k\|_2)(\|q_i^k\|_2) = w_i^k \|q_i^k\|_2^2 \to 0$ for each $i \in \mathcal{I}_0$. In particular, since $w_0^k := 1$ for all $k \in \mathbb{N}$, this implies that $q_0^k \to 0$, or equivalently that $H(x^{k+1}-x^k) \to 0$. In addition, since for each $i \in \mathcal{I}$ both sequences $\{\|q_i^k\|_2\}$ and $\{w_i^k \|q_i^k\|_2\}$ cannot be bounded away from 0, there is a subsequence $\hat{S} \subset \mathbb{N}$ and a partition $\{\mathcal{I}_1, \mathcal{I}_2\}$ of $\mathcal{I}$ such that $\|q_i^k\|_2 \xrightarrow{\hat{S}} 0$ for all $i \in \mathcal{I}_1$ and $w_i^k \|q_i^k\|_2 \xrightarrow{\hat{S}} 0$ for all $i \in \mathcal{I}_2$. Hence, there exists $k_0 \in \hat{S}$ such that for all $k \ge k_0$ we have

$$\begin{aligned} \|q_i^k\|_2 &\le M\Big[\|r_i^k\|_2^2 + \bar{\epsilon}_i^2\Big]^{\frac{1}{2}+\gamma} && \forall\, i \in \mathcal{I}_1 \\ \text{and}\quad w_i^k \|q_i^k\|_2 &\le M\Big[\|r_i^k\|_2^2 + \bar{\epsilon}_i^2\Big]^{\gamma} && \forall\, i \in \mathcal{I}_2. \end{aligned}$$

Therefore, since $w_i^k = (\|r_i^k\|^2 + (\epsilon_i^k)^2)^{-1/2}$, we have for all $k_0 \le k \in \hat{S}$ that

$$\left\|q_i^k\right\|_2 \le M\Big[\left\|r_i^k\right\|_2^2 + \bar{\epsilon}_i^2\Big]^{\frac{1}{2}+\gamma} \quad \forall i \in \mathcal{I}.$$

However, for every such k, Step 2 of the algorithm chooses $\epsilon^{k+1} \in (0, \eta\epsilon^k]$. This contradicts the supposition that $\epsilon^k = \bar{\epsilon} > 0$ for all $k \ge \bar{k}$, so we conclude that $\epsilon^k \to 0$.

(3) It has just been shown in Part (2) that $w_0^k q_0^k = q_0^k \to 0$, so we need only show that $w_i^k \|q_i^k\|_2 \xrightarrow{S} 0$ for each $i \in \mathcal{I}$.

Our first step is to show that for every subsequence $\hat{S} \subset S$ and $i_0 \in \mathcal{I}$, there is a further subsequence $\tilde{S} \subset \hat{S}$ such that $w_{i_0}^k \|q_{i_0}^k\|_2 \xrightarrow{\tilde{S}} 0$. The proof uses a trick from the proof of Part (2). Let $\hat{S} \subset S$ be a subsequence and $i_0 \in \mathcal{I}$. Part (1) implies that $(w_i^k \|q_i^k\|_2)(\|q_i^k\|_2) = w_i^k \|q_i^k\|_2^2 \to 0$ for each $i \in \mathcal{I}_0$. As in the proof of Part (2), this implies that there is a further subsequence $\tilde{S} \subset \hat{S}$ and a partition $\{\mathcal{I}_1, \mathcal{I}_2\}$ of $\mathcal{I}$ such that $\|q_i^k\|_2 \xrightarrow{\tilde{S}} 0$ for all $i \in \mathcal{I}_1$ and $w_i^k \|q_i^k\|_2 \xrightarrow{\tilde{S}} 0$ for all $i \in \mathcal{I}_2$. If $i_0 \in \mathcal{I}_2$, then we would be done, so let us assume that $i_0 \in \mathcal{I}_1$. We can assume that $\tilde{S}$ contains no subsequence on which $w_{i_0}^k \left\| q_{i_0}^k \right\|_2$ converges to 0 since, otherwise, again we would be done. Hence, we assume that $w_{i_0}^k \left\| q_{i_0}^k \right\|_2 \overset{\tilde{S}}{\nrightarrow} 0$. Since $\left\| q_{i_0}^k \right\|_2 \overset{\tilde{S}}{\to} 0$ as $i_0 \in \mathcal{I}_1$, this implies that there is a subsequence $\tilde{S}_0 \subset \tilde{S}$ such that $w_{i_0}^k \overset{\tilde{S}_0}{\to} \infty$, i.e., $(\left\| r_{i_0}^k \right\|_2^2 + (\epsilon_{i_0}^k)^2) \overset{\tilde{S}_0}{\to} 0$. But, by Step 2 of the algorithm, for all $k \in S$,

$$\left\| q_i^k \right\|_2 \leq M \left[\left\| r_i^k \right\|_2^2 + (\epsilon_i^k)^2 \right]^{\frac{1}{2}+\gamma} \quad \forall\, i \in \mathcal{I},$$

or, equivalently,

$$w_i^k \left\| q_i^k \right\|_2 \leq M \left[\left\| r_i^k \right\|_2^2 + (\epsilon_i^k)^2 \right]^{\gamma} \quad \forall\, i \in \mathcal{I},$$

giving the contradiction $w_{i_0}^k \left\| q_{i_0}^k \right\|_2 \overset{\tilde{S}_0}{\to} 0$. Hence, $w_{i_0}^k \left\| q_{i_0}^k \right\|_2 \overset{\tilde{S}}{\to} 0$, and we have shown that for every subsequence $\hat{S} \subset S$ and $i_0 \in \mathcal{I}$, there is $\tilde{S} \subset \hat{S}$ such that $w_i^k \left\| q_i^k \right\|_2 \overset{\tilde{S}}{\to} 0$.

Now, if $W_k q^k \overset{S}{\nrightarrow} 0$, then there would exist a subsequence $\hat{S} \subset S$ and an index $i \in \mathcal{I}$ such that $\{w_i^k \left\| q_i^k \right\|_2\}_{k \in \hat{S}}$ remains bounded away from 0. But, by what we have just shown in the previous paragraph, $\hat{S}$ contains a further subsequence $\tilde{S} \subset \hat{S}$ with $w_i^k \left\| q_i^k \right\|_2 \overset{\tilde{S}}{\to} 0$. This contradiction establishes the result.

(4) By Theorem 4.2.1, we have

$$r_i^k \in N\left(P_{C_i}(A_i x^k + b_i) \,|\, C_i \right) \quad \forall\, i \in \mathcal{I}_0,\ k \in \mathbb{N},$$

from which the result follows.

(5) By convexity, the condition $x^{k+1} \in \operatorname{argmin}_{x \in X} \hat{G}_{(x^k, \epsilon^k)}(x)$ is equivalent to

$$\begin{aligned} 0 &\in \nabla_x \hat{G}_{(x^k,\epsilon^k)}(x^{k+1}) + N\left(x^{k+1} \,|\, X\right) \\ &= g + \sum_{i \in \mathcal{I}_0} A_i^T w_i^k (q_i^k + r_i^k) + N\left(x^{k+1} \,|\, X\right) \\ &= \tilde{A}^T W_k q^k + \nabla \varphi(x^k) + \sum_{i \in \mathcal{I}} A_i^T w_i^k r_i^k + N\left(x^{k+1} \,|\, X\right). \end{aligned}$$

(6) Let $i \in \mathcal{I}$. We know from Part (3) that $w_i^k \left\| q_i^k \right\|_2 \overset{S}{\to} 0$. If $\left\| q_i^k \right\|_2 \overset{S}{\nrightarrow} 0$, then there exists a subsequence $\hat{S} \subset S$ such that $\{\left\| q_i^k \right\|_2\}_{k \in \hat{S}}$ is bounded away from 0, which would

imply that $(\|r_i^k\|_2^2 + (\epsilon_i^k)^2)^{-1/2} = w_i^k \overset{\hat{S}}{\to} 0$. But then $\|r_i^k\|_2 \overset{\hat{S}}{\to} \infty$ since $0 \le \epsilon^k \le \epsilon^0$, which contradicts the boundedness of $\{\text{dist}\left(Ax^k + b \mid C\right)\}_{k\in S}$. □

In the next result, we give conditions under which every cluster point of the subsequence $\{x^k\}_{k\in S}$ is a solution to $\min_{x\in X} J_0(x)$, where S is defined in Lemma 4.2.2. Since J_0 is convex, this is equivalent to showing that $0 \in \partial J_0(\bar{x}) + N(\bar{x} \mid X)$.

Theorem 4.2.10. *Suppose that the sequence $\{(x^k, \epsilon^k)\}$ is generated by IRWA with initial point $x^0 \in X$ and relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$, and that the sequence $\{J(x^k, \epsilon^k)\}$ is bounded below. Let S be defined as in Lemma 4.2.2. If either*

(a) $\ker(A) \cap \ker(H) = \{0\}$ *and* $\{\text{dist}\left(Ax^k + b \mid C\right)\}_{k\in S}$ *is bounded, or*

(b) $X = \mathbb{R}^n$,

then any cluster point $\bar{x}$ of the subsequence $\{x^k\}_{k\in S}$ satisfies $0 \in \partial J_0(\bar{x}) + N(\bar{x} \mid X)$. Moreover, if (a) holds, then $(x^{k+1} - x^k) \overset{S}{\to} 0$.

Proof. Let the sequences $\{q^k\}$, $\{r^k\}$ and $\{W_k\}$ be defined as in Lemma 4.2.2, and let $\bar{x}$ be a cluster point of the subsequence $\{x^k\}_{k\in S}$. Let $\hat{S} \subset S$ be a subsequence such that $x^k \overset{\hat{S}}{\to} \bar{x}$. Without loss of generality, due to the upper semi-continuity of the normal cone operator, the continuity of the projection operator and Lemma 4.2.2(4), we can assume that for each $i \in \mathcal{A}(A\bar{x} + b)$ there exists

$$\bar{u}_i \in \mathbb{B}_2 \cap N(A_i\bar{x} + b_i \mid C_i) \quad \text{such that} \quad w_i^k r_i^k \overset{\hat{S}}{\to} \bar{u}_i. \tag{4.2.15}$$

Also due to the continuity of the projection operator, for each $i \notin I(A\bar{x} + b)$ we have

$$w_i^k r_i^k \overset{\hat{S}}{\to} \frac{(I - P_{C_i})(A_i\bar{x} + b_i)}{\|(I - P_{C_i})(A_i\bar{x} + b_i)\|_2}. \tag{4.2.16}$$

Let us first suppose that (b) holds, i.e., that $X = \mathbb{R}^n$ so that $N(x \mid X) = \{0\}$ for all

$x \in \mathbb{R}^n$. By (4.2.15)-(4.2.16), Lemma 4.2.2 Parts (3) and (5), and (4.2.6), we have

$$0 \in \nabla\varphi(\bar{x}) + \sum_{i \notin \mathcal{A}(A\bar{x}+b)} A_i^T \frac{(I - P_{C_i})(A_i\bar{x} + b_i)}{\|(I - P_{C_i})(A_i\bar{x} + b_i)\|_2} + \sum_{i \in \mathcal{A}(A\bar{x}+b)} A_i^T (\mathbb{B}_2 \cap N(A_i\bar{x} + b_i \,|\, C_i))$$
$$= \partial J_0(\bar{x}).$$

Next, suppose that (a) holds, i.e., that $\ker(A) \cap \ker(H) = \{0\}$ and the set $\{\mathrm{dist}\left(Ax^k + b \mid C\right)\}_{k\in S}$ is bounded. This latter fact and Lemma 4.2.2(6) implies that $q^k \xrightarrow{S} 0$. We now show that $(x^{k+1} - x^k) \xrightarrow{S} 0$. Indeed, if this were not the case, then there would exist a subsequence $\hat{S} \subset S$ and a vector $\bar{w} \in \mathbb{R}^n$ with $\|\bar{w}\|_2 = 1$ such that $\{\|x^{k+1} - x^k\|_2\}_{\hat{S}}$ is bounded away from 0 while $\frac{x^{k+1}-x^k}{\|x^{k+1}-x^k\|_2} \xrightarrow{\hat{S}} \bar{w}$. But then $q^k / \|x^{k+1} - x^k\|_2 \xrightarrow{\hat{S}} 0$ while $q^k / \|x^{k+1} - x^k\|_2 = \tilde{A}\frac{x^{k+1}-x^k}{\|x^{k+1}-x^k\|_2} \xrightarrow{\hat{S}} \tilde{A}\bar{w}$, where $\tilde{A}$ is defined in (4.2.2). But then $0 \neq \bar{w} \in \ker(H) \cap \ker(A) = \ker(\tilde{A})$, a contradiction. Hence, $(x^{k+1} - x^k) \xrightarrow{S} 0$, and so $x^{k+1} = x^k + (x^{k+1} - x^k) \xrightarrow{S} \bar{x}$. In particular, this and the upper semi-continuity of the normal cone operator imply that $\limsup_{k\in S} N\left(x^{k+1} \,|\, X\right) \subset N(\bar{x} \,|\, X)$. Hence, by (4.2.15)–(4.2.16), Lemma 4.2.2 Parts (3) and (5), and (4.2.6), we have

$$0 \in \nabla\varphi(\bar{x}) + \sum_{i \notin I(A\bar{x}+b)} A_i^T \frac{(I - P_{C_i})(A_i\bar{x} + b_i)}{\|(I - P_{C_i})(A_i\bar{x} + b_i)\|_2} + \sum_{i \in I(A\bar{x}+b)} A_i^T (\mathbb{B}_2 \cap N(A_i\bar{x} + b_i \,|\, C_i))$$
$$+ N(\bar{x} \,|\, X)$$
$$= \partial J_0(\bar{x}) + N(\bar{x} \,|\, X),$$

as desired. □

The previously stated Corollary 4.2.8 provides conditions under which the sequence $\{x^k\}$ has cluster points. One of these conditions is that H is positive definite. In such cases, the function J_0 is strongly convex and so the problem (4.1.6) has a unique global solution x^*, meaning that the entire sequence converges to x^*. We formalize this conclusion with the following theorem.

Theorem 4.2.11. *Suppose that H is positive definite and the sequence $\{(x^k, \epsilon^k)\}$ is generated by IRWA with initial point $x^0 \in X$ and relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$. Then, the* *problem* (4.1.6) *has a unique global solution* x^* *and* $x^k \to x^*$*.*

Proof. Since H is positive definite, the function $J(x, \epsilon)$ is strongly convex in x for all $\epsilon \in \mathbb{R}^l_+$. In particular, J_0 is strongly convex and so (4.1.6) has a unique global solution x^*. By Corollary 4.2.8, the set $L(J(x^0, \epsilon^0), \epsilon^0)$ is compact, and, by Lemma 4.2.1, the sequence $J(x^k, \epsilon^k)$ is decreasing; hence, $\{x^k\} \subset L(J(x^0, \epsilon^0), \epsilon^0)$. Therefore, the set $\{\text{dist}\left(Ax^k + b \mid C\right)\}_{k \in S}$ is bounded and $\ker(H) \cap \ker(A) \subset \ker(H) = \{0\}$, and so, by Theorem 4.2.10, the subsequence $\{x^k\}_{k \in S}$ has a cluster point $\bar{x}$ satisfying $0 \in \partial J_0(\bar{x}) + N(\bar{x} \mid X)$. But the only such point is $\bar{x} = x^*$, and hence $x^k \overset{S}{\to} x^*$.

Since the sequence $\{J(x^k, \epsilon^k)\}$ is monotonically decreasing and bounded below by Corollary 4.2.8, it has a limit $\tilde{J}$. Since $x^k \overset{S}{\to} x^*$, we have $\tilde{J} = \min_{x \in X} J_0(x)$. Let $\tilde{S}$ be any subsequence of $\mathbb{N}$. Since $\{x^k\}_{k \in \tilde{S}} \subset L(J(x^0, \epsilon^0), \epsilon^0)$ (which is compact by Corollary 4.2.8(2)), this subsequence has a further subsequence $\tilde{S}_0 \subset \tilde{S}$ such that $x^k \overset{\tilde{S}_0}{\to} \bar{x}$ for some $\bar{x} \in X$. For this subsequence, $J(x^k, \epsilon^0) \overset{\tilde{S}_0}{\to} \tilde{J}$, and, by continuity, $J(x^k, \epsilon^0) \overset{\tilde{S}_0}{\to} J(\bar{x}, 0) = J_0(\bar{x})$. Hence, $\bar{x} = x^*$ by uniqueness. Therefore, since every subsequence of $\{x^k\}$ has a further subsequence that converges to x^*, it must be the case that the entire sequence converges to x^*. □

4.2.4 Complexity of IRWA

A point $\tilde{x} \in X$ is an ε-optimal solution to (4.1.6) if

$$J_0(\tilde{x}) \leq \inf_{x \in X} J_0(x) + \varepsilon. \tag{4.2.17}$$

In this section, we prove the following result.

Theorem 4.2.12. *Consider the problem* (4.1.6) *with* $X = \mathbb{R}^n$ *and* H *positive definite. Let* $\varepsilon > 0$ *and* $\epsilon \in \mathbb{R}^l_{++}$ *be such that*

$$\|\epsilon\|_1 \leq \varepsilon/2 \quad \textit{and} \quad \varepsilon \leq 4l\tilde{\varepsilon}, \tag{4.2.18}$$

where $\tilde{\varepsilon} := \min_{i \in \mathcal{I}} \epsilon_i$*. Suppose that the sequence* $\{(x^k, \epsilon^k)\}$ *is generated by IRWA with* *initial point* $x^0 \in \mathbb{R}^n$ *and relaxation vector* $\epsilon^0 = \epsilon \in \mathbb{R}^l_{++}$, *and that the relaxation vector is kept fixed so that* $\epsilon^k = \epsilon$ *for all* $k \in \mathbb{N}$. *Then, in at most* $O(1/\varepsilon^2)$ *iterations,* x^k *is an* ε*-optimal solution to* (4.1.6), *i.e.,* (4.2.17) *holds with* $\tilde{x} = x^k$.

The proof of this result requires a few preliminary lemmas. For ease of presentation, we assume that the hypotheses of Theorem 4.2.12 hold throughout this section. Thus, in particular, Corollary 4.2.8 and the strict convexity and coercivity of J tells us that there exists $\tau > 0$ such that

$$\left\| x^k - x^\epsilon \right\|_2 \leq \tau \quad \text{for all } k \in \mathbb{N}, \tag{4.2.19}$$

where x^ϵ is the solution to $\min_{x\in\mathbb{R}^n} J(x,\epsilon)$. Let w_i for $i \in \mathcal{I}$ and $\tilde{A}$ be given as in (4.2.1) and (4.2.2), respectively. In addition, define

$$R_i(r_i) := \frac{r_i}{\sqrt{\|r_i\|_2^2 + \epsilon_i^2}}, \quad r_i(x) := (I - P_{C_i})(A_i x + b_i) \text{ for } i \in \mathcal{I}$$
$$\text{and} \quad u(x,\epsilon) := \nabla\varphi(x) + \sum_{i\in\mathcal{I}} w_i(x,\epsilon) A_i^T r_i(x).$$

Recall that

$$\partial_x J(x,\epsilon)$$
$$= \nabla\varphi(x) + \sum_{i\notin\mathcal{A}(Ax+b)} w_i(x,\epsilon)A_i^T r_i(x) + \sum_{i\in\mathcal{A}(Ax+b)} w_i(x,\epsilon)A_i^T(\mathbb{B}_2 \cap N(A_i x + b_i|C_i)),$$

so that $u(x,\epsilon) \in \partial_x J(x,\epsilon)$. It is straightforward to show that, for each $i \in \mathcal{I}$, we have

$$\nabla_{r_i} R_i(r_i) = \frac{1}{\sqrt{\|r_i\|_2^2 + \epsilon_i^2}} \left(I - \frac{r_i r_i^T}{\|r_i\|_2^2 + \epsilon_i^2} \right)$$

so that

$$\|\nabla_{r_i} R_i(r_i)\|_2 \leq 1/\epsilon_i \quad \forall\, r_i. \tag{4.2.20}$$

Consequently, for each $i \in \mathcal{I}$, the function R_i is globally Lipschitz continuous with Lipschitz constant $1/\epsilon_i$. This allows us to establish a similar result for the mapping $u(x,\epsilon)$ as a function of x, which we prove as our next result. For convenience, we use

$$\bar{u} := u(\bar{x}, \epsilon), \;\; \hat{u} := u(\hat{x}, \epsilon), \;\text{ and } u^k := u(x^k, \epsilon),$$

and similar shorthand for $w_i(x, \epsilon_i)$, $W(x, \epsilon)$, and $r_i(x)$.

Lemma 4.2.3. *Let the hypotheses of Theorem 4.2.12 hold. Moreover, let λ be the largest eigenvalue of H and σ_1 be an upper bound on all singular values of the matrices A_i for $i \in \mathcal{I}$. Then, as a function of x, the mapping $u(x, \epsilon)$ is globally Lipschitz continuous with Lipschitz constant $\beta := \lambda + l\sigma_1^2/\tilde{\varepsilon}$.*

Proof. By Theorem 4.2.1, for all $\{\bar{x}, \hat{x}\} \subset \mathbb{R}^n$, we have

$$\|\bar{r}_i - \hat{r}_i\|_2 \le \|A_i(\bar{x} - \hat{x})\|_2 \le \sigma_1 \|\bar{x} - \hat{x}\|_2 . \tag{4.2.21}$$

Therefore,

$$\begin{aligned} \|\bar{u} - \hat{u}\|_2 &= \left\| H(\bar{x} - \hat{x}) + \sum_{i \in \mathcal{I}} A_i^T (R_i(\bar{r}_i) - R_i(\hat{r}_i)) \right\|_2 \\ &\le \|H\|_2 \|\bar{x} - \hat{x}\|_2 + \tfrac{1}{\tilde{\epsilon}} \sum_{i \in \mathcal{I}} \|A_i\|_2 \|\bar{r}_i - \hat{r}_i\|_2 \\ &\le \|H\|_2 \|\bar{x} - \hat{x}\|_2 + \tfrac{1}{\tilde{\epsilon}} \sum_{i \in \mathcal{I}} \|A_i\|_2^2 \|\bar{x} - \hat{x}\|_2 \\ &\le (\lambda + l\sigma_1^2/\tilde{\varepsilon}) \|\bar{x} - \hat{x}\|_2 , \end{aligned}$$

where the first inequality follows from (4.2.20), the second from (4.2.21), and the last from the fact that the 2-norm of a matrix equals its largest singular value. □

By Lemma 4.2.3 and the subgradient inequality, we obtain the bound

$$0 \le J(\bar{x}, \epsilon) - J(\hat{x}, \epsilon) - \langle \hat{u}, \, \bar{x} - \hat{x} \rangle \le \langle \bar{u} - \hat{u}, \, \bar{x} - \hat{x} \rangle \le \beta \|\bar{x} - \hat{x}\|_2^2 . \tag{4.2.22}$$

Moreover, by Part (5) of Lemma 4.2.2, we have

$$-\tilde{A}^T W_k q^k = -\tilde{A}^T W_k \tilde{A}(x^{k+1} - x^k) = u^k \in \partial_x J(x^k, \epsilon).$$

If we now define $D_k := \tilde{A}^T W_k \tilde{A}$, then $x^k - x^{k+1} = D_k^{-1} u^k$ and

$$(q^k)^T W_k q^k = (x^k - x^{k+1})^T D_k (x^k - x^{k+1}) = (u^k)^T D_k^{-1} u^k. \tag{4.2.23}$$

This gives the following bound on the decrease in J when going from x^k to x^{k+1}.

Lemma 4.2.4. *Let the hypotheses of Lemma 4.2.3 hold. Then,*

$$J(x^{k+1}, \epsilon) - J(x^k, \epsilon) \leq -\alpha \|u^k\|_2^2,$$

where $\alpha := \tilde{\varepsilon}/(2\sigma_0^2)$ *with* σ_0 *the largest singular value of* $\tilde{A}$.

Proof. By Lemma 4.2.1 and (4.2.23), we have

$$J(x^{k+1}, \epsilon) - J(x^k, \epsilon) \leq -\tfrac{1}{2}(q^k)^T W_k q^k = -\tfrac{1}{2}(u^k)^T D_k^{-1} u^k.$$

Since the $\|D_k\|_2 \leq \|W_k^{1/2}\|_2^2 \|\tilde{A}\|_2^2$, we have that the largest eigenvalue of D_k is bounded above by $\sigma_0^2/\tilde{\varepsilon}$. This implies $\frac{1}{2}(u^k)^T D_k^{-1} u^k \geq \alpha \left\|u^k\right\|_2^2$, which gives the result. □

The following theorem is the main tool for proving Theorem 4.2.12.

Theorem 4.2.13. *Let the hypotheses of Lemma 4.2.4 hold, and, as in* (4.2.19)*, let* x^ϵ *be the solution to* $\min_{x \in \mathbb{R}^n} J(x, \epsilon)$*. Then,*

$$J(x^k, \epsilon) - J(x^\epsilon, \epsilon) \leq \frac{32 l^2 \sigma_0^2 \tau^2}{k\varepsilon} \left[\frac{\|u^\epsilon\|_2 \varepsilon + \tau(\lambda\varepsilon + l\sigma_1^2)}{\|u^\epsilon\|_2 \varepsilon + \tau(\lambda\varepsilon + 4l^2\sigma_1^2) + 8l\tau\sigma_0^2/k} \right]. \tag{4.2.24}$$

Therefore, IRWA requires $O(1/\varepsilon^2)$ *iterations to reach* ε*-optimality for* $J(x, \epsilon)$*, i.e.,*

$$J(x^k, \epsilon) - J(x^\epsilon, \epsilon) \leq \varepsilon.$$

Proof. Set $\delta^j := J(x^j, \epsilon) - J(x^\epsilon, \epsilon)$ for all $j \in \mathbb{N}$. Then, by Lemma 4.2.4,

$$\begin{aligned} 0 \le \delta^{j+1} &= J(x^{j+1}, \epsilon) - J(x^\epsilon, \epsilon) \\ &\le J(x^j, \epsilon) - J(x^\epsilon, \epsilon) - \alpha \|u^j\|_2^2 = \delta^j - \alpha \|u^j\|_2^2 \le \delta^j. \end{aligned} \tag{4.2.25}$$

If for some $j < k$ we have $\delta^j = 0$, then (4.2.25) implies that $\delta^k = 0$ and $u^k = 0$, which in turn implies that $x^{k+1} = x^\epsilon$ and the bound (4.2.24) holds trivially. In the remainder of the proof, we only consider the nontrivial case where $\delta^j > 0$ for $j = 0, ..., k-1$.

Consider $j \in \{0, \dots, k-1\}$. By the convexity of J and (4.2.19), we have

$$\delta^j = J(x^j, \epsilon) - J(x^\epsilon, \epsilon) \le (u^j)^T (x^j - x^\epsilon) \le \|u^j\|_2 \|x^j - x^\epsilon\| \le \tau \|u^j\|_2.$$

Combining this with (4.2.25), gives

$$\delta^{j+1} \le \delta^j - \frac{\alpha}{\tau^2} (\delta^j)^2.$$

Dividing both sides by $\delta^{j+1}\delta^j$ and noting that $\frac{\delta^j}{\delta^{j+1}} \ge 1$ yields

$$\frac{1}{\delta^{j+1}} - \frac{1}{\delta^j} \ge \frac{\alpha}{\tau^2} \frac{\delta^j}{\delta^{j+1}} \ge \frac{\alpha}{\tau^2}. \tag{4.2.26}$$

Summing both sides of (4.2.26) from 0 to $k-1$, we obtain

$$\frac{1}{\delta^k} \ge \frac{\alpha k}{\tau^2} + \frac{1}{\delta^0} = \frac{\alpha \delta^0 k + \tau^2}{\delta^0 \tau^2}, \tag{4.2.27}$$

or, equivalently,

$$\delta^k \le \frac{\delta^0 \tau^2}{\alpha \delta^0 k + \tau^2}. \tag{4.2.28}$$

The inequality (4.2.22) implies that

$$\delta^0 = J(x^0, \epsilon) - J(x^\epsilon, \epsilon) \le (u^\epsilon)^T (x^0 - x^\epsilon) + \beta \|x^0 - x^\epsilon\|_2^2 \le \tau(\|u^\epsilon\|_2 + \beta\tau),$$

which, together with (4.2.27), implies that

$$\frac{\alpha\delta^0 k + \tau^2}{\delta^0\tau^2} \geq \frac{\alpha k}{\tau^2} + \frac{1}{\tau(\|u^\epsilon\|_2 + \beta\tau)}.$$

Rearranging, one has

$$\frac{\tau^2\delta^0}{\alpha k\delta^0 + \tau^2} \leq \frac{\tau^2(\|u^\epsilon\|_2 + \beta\tau)}{\alpha k(\|u^\epsilon\|_2 + \beta\tau) + \tau}.$$

Substituting in $\beta = \lambda + l\sigma_1^2/\tilde{\varepsilon}$ and $\alpha = \tilde{\varepsilon}/(2\sigma_0^2)$ defined in Lemmas 4.2.3 and 4.2.4, respectively, and then combining with (4.2.28) gives

$$\delta^k \leq \frac{\tau^2(\|u^\epsilon\|_2 + \tau(\lambda + l\sigma_1^2/\tilde{\varepsilon}))}{(\tilde{\varepsilon}/(2\sigma_0^2))k(\|u^\epsilon\|_2 + \tau(\lambda + l\sigma_1^2/\tilde{\varepsilon})) + \tau} = \frac{2\sigma_0^2\tau^2}{k\tilde{\varepsilon}}\left[\frac{\|u^\epsilon\|_2\,\tilde{\varepsilon} + \tau(\lambda\tilde{\varepsilon} + l\sigma_1^2)}{\|u^\epsilon\|_2\,\tilde{\varepsilon} + \tau(\lambda\tilde{\varepsilon} + l\sigma_1^2) + 2\tau\sigma_0^2/k}\right].$$

Finally, using the inequalities $\varepsilon \leq 4l\tilde{\varepsilon}$ and $\tilde{\varepsilon} \leq \varepsilon$ (recall (4.2.18)) gives

$$\delta^k \leq \frac{32l^2\sigma_0^2\tau^2}{k\varepsilon}\left[\frac{\|u^\epsilon\|_2\,\varepsilon + \tau(\lambda\varepsilon + l\sigma_1^2)}{\|u^\epsilon\|_2\,\varepsilon + \tau(\lambda\varepsilon + 4l^2\sigma_1^2) + 8l\tau\sigma_0^2/k}\right],$$

which is the desired inequality. □

We can now prove Theorem 4.2.12.

Theorem 4.2.12. Let $x^* = \arg\min_{x\in\mathbb{R}^n} J_0(x)$. Then, by convexity in ϵ,

$$\begin{aligned} J(x^\epsilon, \epsilon) - J(x^*, 0) &\leq [\partial_\epsilon J(x^\epsilon, \epsilon)]^T(\epsilon - 0) \\ &= \sum_{i\in\mathcal{I}} \frac{\epsilon_i^2}{\sqrt{\|r_i(x^\epsilon)\|_2^2 + \epsilon_i^2}} \leq \sum_{i\in\mathcal{I}} \epsilon_i = \|\epsilon\|_1 \leq \varepsilon/2. \end{aligned}$$

By Theorem 4.2.13, IRWA needs $O(1/\varepsilon^2)$ iterations to reach

$$J(x^k, \epsilon) - J(x^\epsilon, \epsilon) \leq \varepsilon/2.$$

Combining these two inequalities yields the result. □

4.3 An Alternating Direction Augmented Lagrangian Algorithm

For comparison with IRWA, we now describe an alternating direction augmented Lagrangian method for solving problem (4.1.6). This approach, like IRWA, can be solved by matrix-free methods. Defining

$$\hat{J}(x,p) := \varphi(x) + \operatorname{dist}(p \mid C),$$

where $\operatorname{dist}(p \mid C)$ is defined as in (4.1.5), the problem (4.1.6) has the equivalent form

$$\min_{x \in X, p} \ \hat{J}(x,p) \text{ subject to } Ax + b = p, \tag{4.3.1}$$

where $p := (p_1^T, \dots, p_l^T)^T$. In particular, note that $J_0(x) = \hat{J}(x, Ax+b)$. Defining dual variables $(u_1, \dots, u_l)$, a partial Lagrangian for (4.3.1) is given by

$$L(x,p,u) := \hat{J}(x,p) + \langle u,\ Ax+b-p \rangle + \delta(x \mid X),$$

and the corresponding augmented Lagrangian, with penalty parameter $\mu > 0$, is

$$L(x,p,u,\mu) := \hat{J}(x,p) + \tfrac{1}{2\mu}\|Ax+b-p+\mu u\|_2^2 - \tfrac{\mu}{2}\|u\|_2^2 + \delta(x \mid X).$$

(Observe that due to their differing numbers of inputs, the Lagrangian value $L(x,p,u)$ and augmented Lagrangian value $L(x,p,u,\mu)$ should not be confused with each other, nor with the level set value $L(\alpha,\epsilon)$ defined in Theorem 4.2.7.)

We now state the algorithm.

Alternating Direction Augmented Lagrangian Algorithm (ADAL)

Step 0: (Initialization) Choose an initial point $x^0 \in X$, dual vectors $u_i^0 \in \mathbb{R}^{m_i}$ for $i \in \mathcal{I}$, and penalty parameter $\mu > 0$. Let $\sigma \geq 0$ and $\sigma'' \geq 0$ be two scalars which serve as termination tolerances for the stepsize and constraint residual, respectively. Set $k := 0$.

Step 1: (Solve the augmented Lagrangian subproblems for (x^{k+1}, p^{k+1}))

Compute a solution p^{k+1} to the problem

$$\mathcal{L}_p(x^k, p, u^k, \mu) : \quad \min_p L(x^k, p, u^k, \mu),$$

and a solution x^{k+1} to the problem

$$\mathcal{L}_x(x, p^{k+1}, u^k, \mu) : \quad \min_x L(x, p^{k+1}, u^k, \mu).$$

Step 2: (Set the new multipliers u^{k+1})

Set

$$u^{k+1} := u^k + \tfrac{1}{\mu}(Ax^{k+1} + b - p^{k+1}).$$

Step 3: (Check stopping criteria)

If $\left\|x^{k+1} - x^k\right\|_2 \leq \sigma$ and $\left\|Ax^{k+1} + b - p^{k+1}\right\|_* \leq \sigma''$, then stop; else, set $k := k+1$ and go to Step 1.

Remark 4.3.1. As for IRWA, one can also base the stopping criteria of Step 3 on a percent reduction in duality gap; recall Remark 4.2.4.

4.3.1 Properties of $\mathcal{L}_p(x, p, u, \mu)$ and $\mathcal{L}_x(x, p, u, \mu)$

Before addressing the convergence properties of the ADAL algorithm, we discuss properties of the solutions to the subproblems $\mathcal{L}_p(x, p, u, \mu)$ and $\mathcal{L}_x(x, p, u, \mu)$.

The subproblem $\mathcal{L}_p(x^k, p, u^k, \mu)$ is separable. Defining

$$s_i^k := A_i x^k + b_i + \mu u_i^k \quad \forall\, i \in \mathcal{I},$$

the solution of $\mathcal{L}_p(x^k, p, u^k, \mu)$ can be written explicitly, for each $i \in \mathcal{I}$, as

$$p_i^{k+1} := \begin{cases} P_{C_i}(s_i^k) & \text{if } \operatorname{dist}_2\left(s_i^k \mid C_i\right) \le \mu \\ s_i^k - \frac{\mu}{\operatorname{dist}_2\left(s_i^k \mid C_i\right)}(s_i^k - P_{C_i}(s_i^k)) & \text{if } \operatorname{dist}_2\left(s_i^k \mid C_i\right) > \mu. \end{cases} \tag{4.3.2}$$

Subproblem $\mathcal{L}_x(x, p^{k+1}, u^k, \mu)$, on the other hand, involves the minimization of a convex quadratic over X, which can be solved by matrix-free methods.

Along with the dual variable estimates $\{u_i^k\}$, we define the auxiliary estimates

$$\hat{u}^{k+1} := u^{k+1} - \tfrac{1}{\mu} q^k, \quad \text{where} \quad q^k := A(x^{k+1} - x^k) \quad \text{as in IRWA Step 2.}$$

First-order optimality conditions for (4.3.1) are then given by

$$0 \in \partial \operatorname{dist}\left(p \mid C\right) - u, \tag{4.3.3a}$$

$$0 \in \nabla\varphi(x) + A^T u + N(x|X), \tag{4.3.3b}$$

$$0 = Ax + b - p, \tag{4.3.3c}$$

or, equivalently,

$$0 \in \partial J_0(x) = \nabla\varphi(x) + A^T \partial \operatorname{dist}\left(\cdot \mid C\right)(Ax + b) + N\left(x \mid X\right).$$

The next lemma relates the iterates and these optimality conditions.

Lemma 4.3.1. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, for all $k \in \mathbb{N}$, we have*

$$\hat{u}^{k+1} \in \partial \operatorname{dist}\left(p^{k+1} \mid C\right) \text{ and } -\tfrac{1}{\mu} A^T q^k \in \nabla\varphi(x^{k+1}) + A^T \hat{u}^{k+1} + N\left(x^{k+1} \mid X\right). \tag{4.3.4}$$

Therefore,

$$-\tfrac{1}{\mu} A^T q^k \in \nabla\varphi(x^{k+1}) + A^T \partial \operatorname{dist}\left(p^{k+1} \mid C\right) + N\left(x^{k+1} \mid X\right).$$

Moreover, for all $k \geq 1$, we have

$$\left\|\hat{u}^k\right\|_* \leq 1, \quad \left\|s^k\right\|_* \leq \mu, \quad \text{and} \quad \left\|p^k\right\|_* \leq \hat{\mu}, \tag{4.3.5}$$

where $\hat{\mu} := \max\{\mu, \sup_{\|s\|_* \leq \mu} \|P_C(s)\|_*\} < \infty$.

Proof. By ADAL Step 1, the auxiliary variable p^{k+1} satisfies

$$0 \in \partial\text{dist}\left(p^{k+1} \,|\, C\right) - u^k - \tfrac{1}{\mu}(Ax^k + b - p^{k+1}),$$

which, along with ADAL Step 2, implies that

$$\begin{aligned} u^{k+1} &\in \partial\text{dist}\left(p^{k+1} \,|\, C\right) + \tfrac{1}{\mu}(Ax^{k+1} + b - p^{k+1}) - \frac{1}{\mu}(Ax^k + b - p^{k+1}) \\ &= \partial\text{dist}\left(p^{k+1} \,|\, C\right) + \tfrac{1}{\mu}q^k. \end{aligned}$$

Hence, the first part of (4.3.4) holds. Then, again by ADAL Step 1, x^{k+1} satisfies

$$0 \in \nabla\varphi(x^{k+1}) + \frac{1}{\mu}A^T(Ax^{k+1} + b - p^{k+1} + \mu u^k) + N(x^{k+1}|X).$$

which, along with ADAL Step 2, implies that

$$0 \in \nabla\varphi(x^{k+1}) + A^T u^{k+1} + N(x^{k+1}|X). \tag{4.3.6}$$

Hence, the second part of (4.3.4) holds.

The first bound in (4.3.5) follows from the first part of (4.3.4). The second bound in (4.3.5) follows from the first bound and the fact that for $k \in \mathbb{N}$ we have

$$s^k = Ax^k + b + \mu u^k = \mu u^{k+1} - q^k = \mu \hat{u}^{k+1}.$$

As for the third bound, note that if, for some $i \in \mathcal{I}$, we have $\text{dist}_2\left(s_i^{k-1} \,|\, C_i\right) \leq \mu$, then, by (4.3.2), we have $\|p_i^k\|_2 \leq \hat{\mu}$; on the other hand, if $\text{dist}_2\left(s_i^{k-1} \,|\, C_i\right) > \mu$ so that

$0 < \xi := \mu/\mathrm{dist}_2\left(s_i^{k-1} \mid C_i\right) < 1$, then, by (4.3.2) and the second bound in (4.3.5),

$$\left\|p_i^k\right\|_2 \leq (1-\xi)\left\|s_i^{k-1}\right\|_2 + \xi\hat{\mu} \leq \hat{\mu}.$$

Consequently, $\left\|p^k\right\|_* = \sup_{i\in\mathcal{I}}\left\|p_i^k\right\|_2 \leq \hat{\mu}$. □

For the remainder of our discussion of ADAL, we define the residuals

$$z^{k+1} := Ax^{k+1} + b - p^{k+1}.$$

Lemma 4.3.1 tells us that the deviation of $(p^{k+1}, \hat{u}^{k+1})$ from satisfying the first-order optimality conditions for (4.3.3) can be measured by

$$E^{k+1} = \max\{\left\|q^k\right\|, \left\|z^{k+1}\right\|_*\}. \tag{4.3.7}$$

4.3.2 Convergence of ADAL

In this section, we establish the global convergence properties of the ADAL algorithm. The proofs in this section are standard for algorithms of this type (e.g., see [5]), but we include them for the sake of completeness. We make use of the following standard assumption.

Assumption 4.3.2. *There exists a point* (x^*, p^*, u^*) *satisfying* (4.3.3).

Since (4.3.1) is convex, this assumption is equivalent to the existence of a minimizer. Notice that (x^*, p^*) is a minimizer of the convex function $L(x, p, u^*)$ over X. We begin our analysis by providing useful bounds on the optimal primal objective value.

Lemma 4.3.2. *Suppose that the sequence* $\{(x^k, p^k, u^k)\}$ *is generated by ADAL with initial point* $x^0 \in X$*. Then, under Assumption 4.3.2, we have for all* $k \in \mathbb{N}$ *that*

$$(u^*)^T z^{k+1} \geq \hat{J}(x^*, p^*) - \hat{J}(x^{k+1}, p^{k+1}) \geq (u^{k+1})^T z^{k+1} - \frac{1}{\mu}(q^k)^T(p^* - p^{k+1}). \tag{4.3.8}$$

Proof. Since (x^*, p^*, u^*) is a saddle point of L, it follows that $Ax^* + b - p^* = 0$, which implies by the fact that $x^{k+1} \in X$ that

$$\hat{J}(x^*, p^*) = L(x^*, p^*, u^*) \leq L(x^{k+1}, p^{k+1}, u^*).$$

Rearranging, we obtain the first inequality in (4.3.8).

We now show the second inequality in (4.3.8). Recall that Steps 1 and 2 of ADAL tell us that (4.3.6) holds for all $k \in \mathbb{N}$. Therefore, x^{k+1} is first-order optimal for

$$\min_{x \in X} \ \varphi(x) + (u^{k+1})^T Ax.$$

Since this is a convex problem and $x^* \in X$, we have

$$\varphi(x^*) + (u^{k+1})^T Ax^* \geq \varphi(x^{k+1}) + (u^{k+1})^T Ax^{k+1}. \tag{4.3.9}$$

Similarly, by the first expression in (4.3.4), p^{k+1} is first-order optimal for

$$\min_{p} \ \mathrm{dist}\,(p \mid C\,) - (\hat{u}^{k+1})^T p.$$

Hence, by the convexity of this problem, we have

$$\mathrm{dist}\,(p^* \mid C\,) - (\hat{u}^{k+1})^T p^* \geq \mathrm{dist}\left(p^{k+1} \mid C\right) - (\hat{u}^{k+1})^T p^{k+1}. \tag{4.3.10}$$

By adding (4.3.9) and (4.3.10), we obtain

$$\begin{aligned}
&\hat{J}(x^*, p^*) - \hat{J}(x^{k+1}, p^{k+1}) \\
&\geq (\hat{u}^{k+1})^T (p^* - p^{k+1}) + (u^{k+1})^T A(x^{k+1} - x^*) \\
&= (u^{k+1})^T (p^* - p^{k+1}) - \tfrac{1}{\mu}(q^k)^T (p^* - p^{k+1}) + (u^{k+1})^T A(x^{k+1} - x^*) \\
&= (u^{k+1})^T \left((p^* - Ax^*) - b) - (p^{k+1} - Ax^{k+1} - b)\right) - \tfrac{1}{\mu}(q^k)^T (p^* - p^{k+1}) \\
&= (u^{k+1})^T z^{k+1} - \tfrac{1}{\mu}(q^k)^T (p^* - p^{k+1}),
\end{aligned}$$

which completes the proof. $\square$

Consider the distance measure to (x^*, u^*) defined by

$$\omega^k := \tfrac{1}{\mu} \left\| A(x^k - x^*) \right\|_2^2 + \mu \left\| u^k - u^* \right\|_2^2.$$

In our next lemma, we show that this measure decreases monotonically.

Lemma 4.3.3. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, under Assumption 4.3.2 holds, we have for all $k \geq 1$ that*

$$\tfrac{1}{\mu}(\left\| z^{k+1} \right\|_2^2 + \left\| q^k \right\|_2^2) + 2(x^{k+1} - x^k)^T H (x^{k+1} - x^k) \leq \omega^k - \omega^{k+1}. \tag{4.3.11}$$

Proof. By using the extremes of the inequality (4.3.8) and rearranging, we obtain

$$(u^{k+1} - u^*)^T z^{k+1} - \tfrac{1}{\mu}(q^k)^T (p^* - p^{k+1}) \leq 0.$$

Since (x^*, p^*, u^*) is a saddle point of L, and so $Ax^* + b = p^*$, this implies

$$(u^{k+1} - u^*)^T z^{k+1} - \tfrac{1}{\mu}(q^k)^T z^{k+1} + \tfrac{1}{\mu}(x^{k+1} - x^k)^T A^T A (x^{k+1} - x^*) \leq 0. \tag{4.3.12}$$

The update in Step 2 yields $u^{k+1} = u^k + \frac{1}{\mu} z^{k+1}$, so we have

$$(u^{k+1} - u^*)^T z^{k+1} = \left[(u_i^k - u^*)^T z^{k+1} + \tfrac{1}{2\mu} \left\| z^{k+1} \right\|_2^2 \right] + \tfrac{1}{2\mu} \left\| z^{k+1} \right\|_2^2. \tag{4.3.13}$$

Let us now consider the first grouped term in (4.3.13). From ADAL Step 2, we have

$z^{k+1} = \mu(u^{k+1} - u^k)$, which gives

$$\begin{aligned}(u^k - u^*)^T z^{k+1} + \tfrac{1}{2\mu} \left\| z^{k+1} \right\|_2^2 &= \mu(u^k - u^*)^T (u^{k+1} - u^k) + \tfrac{\mu}{2} \|u^{k+1} - u^k\|_2^2 \\ &= \mu(u^k - u^*)^T (u^{k+1} - u^*) - \mu(u^k - u^*)^T (u^k - u^*) \\ &\quad + \tfrac{\mu}{2} \|(u^{k+1} - u^*) - (u^k - u^*)\|_2^2 \\ &= \tfrac{\mu}{2} (\|u^{k+1} - u^*\|_2^2 - \|u^k - u^*\|_2^2). \end{aligned} \tag{4.3.14}$$

Adding the final term $\frac{1}{2\mu} \left\| z^{k+1} \right\|_2^2$ in (4.3.13) to the second and third terms in (4.3.12),

$$\begin{aligned} &\tfrac{1}{\mu} \left(\tfrac{1}{2} \left\| z^{k+1} \right\|_2^2 - (q^k)^T z^{k+1} + (x^{k+1} - x^k)^T A^T A (x^{k+1} - x^*) \right) \\ &= \tfrac{1}{\mu} \left(\tfrac{1}{2} \left\| z^{k+1} \right\|_2^2 - (q^k)^T z^{k+1} + (x^{k+1} - x^k)^T A^T A ((x^{k+1} - x^k) + (x^k - x^*)) \right) \\ &= \tfrac{1}{\mu} \left(\tfrac{1}{2} \left\| z^{k+1} - q^k \right\|_2^2 + \tfrac{1}{2} \|q^k\|_2^2 + (x^{k+1} - x^k)^T A^T A (x^k - x^*) \right) \\ &= \tfrac{1}{\mu} (\tfrac{1}{2} \left\| z^{k+1} - q^k \right\|_2^2 + \tfrac{1}{2} \|A((x^{k+1} - x^*) - (x^k - x^*))\|_2^2 \\ &\quad + ((x^{k+1} - x^*) - (x^k - x^*))^T A^T A (x^k - x^*)) \\ &= \tfrac{1}{2\mu} \left(\left\| z^{k+1} - q^k \right\|_2^2 + \|A(x^{k+1} - x^*)\|_2^2 - \|A(x^k - x^*)\|_2^2 \right) \end{aligned} \tag{4.3.15}$$

From (4.3.13), (4.3.14), and (4.3.15), we have that (4.3.12) reduces to

$$\omega^{k+1} - \omega^k \le -\frac{1}{\mu} \left\| z^{k+1} - q^k \right\|_2^2.$$

Since (4.3.6) holds for $k \ge 1$, we have

$$-(v^{k+1} - v^k) = H(x^{k+1} - x^k) + A^T (u^{k+1} - u^k),$$

for some $v^{k+1} \in N(x^{k+1}|X)$ and $v^k \in N(x^k|X)$. Therefore,

$$\begin{aligned}(u^{k+1} - u^k)^T q^k &= -(v^{k+1} - v^k)^T(x^{k+1} - x^k) - (x^{k+1} - x^k)^T H(x^{k+1} - x^k) \\ &\le -(x^{k+1} - x^k)^T H(x^{k+1} - x^k), \end{aligned} \tag{4.3.16}$$

where the inequality follows since the normal cone operator $N(\cdot\,|\,C)$ is a monotone operator [72]. Using this inequality in the expansion of the right-hand side of (4.3.16) along with the equivalence $z^{k+1} = \mu(u^{k+1} - u^k)$, gives

$$\begin{aligned}\omega^{k+1} - \omega^k &\le -\tfrac{1}{\mu}\left(\|z^{k+1}\|_2^2 - 2\mu(u^{k+1} - u^k)^T q^k + \|q^k\|_2^2\right) \\ &\le -\tfrac{1}{\mu}(\|z^{k+1}\|_2^2 + \|q^k\|_2^2) + 2(u_i^{k+1} - u_i^k)^T q_i^k \\ &\le -\tfrac{1}{\mu}(\|z^{k+1}\|_2^2 + \|q^k\|_2^2) - 2(x^{k+1} - x^k)^T H(x^{k+1} - x^k),\end{aligned}$$

as desired. □

We now state and prove our main convergence theorem for ADAL.

Theorem 4.3.3. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, under Assumption 4.3.2, we have*

$$\lim_{k\to\infty} q^k = 0, \quad \lim_{k\to\infty} z^{k+1} = 0, \quad \textit{and so} \quad \lim_{k\to\infty} E^k = 0.$$

Moreover, the sequences $\{u^k\}$ and $\{Ax^k\}$ are bounded and

$$\lim_{k\to\infty} \hat{J}(x^k, p^k) = \hat{J}(x^*, p^*) = J_0(x^*).$$

Proof. Summing (4.3.11) over all $k \ge 1$ yields

$$\sum_{k=1}^{\infty}\left(2(x^{k+1} - x^k)^T H(x^{k+1} - x^k) + \tfrac{1}{\mu}(\|z^{k+1}\|_2^2 + \|q^k\|_2^2)\right) \le \omega^1,$$

which, since $H \succeq 0$, implies that $z^{k+1} \to 0$ and $q^k \to 0$. Consequently, $E^k \to 0$.

The sequence $\{u^k\}$ is bounded since $\hat{u}^{k+1} + (1/\mu)q^k = u^{k+1}$, where $\{\hat{u}^k\}$ is bounded by (4.3.5) and $q^k \to 0$. Similarly, the sequence $\{Ax^k\}$ is bounded since $\mu(u^{k+1} - u^k) + p^{k+1} - b = Ax^{k+1}$, where the sequence $\{p^k\}$ is bounded by (4.3.5). Finally, by (4.3.8), we have that $\hat{J}(x^k, p^k) \to \hat{J}(x^*, p^*)$ since both $z^k \to 0$ and $q^k \to 0$ while $\{p^k\}$ and $\{u^k\}$ are both bounded. □

Corollary 4.3.4. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, under Assumption 4.3.2, every cluster point of the sequence $\{x^k\}$ is a solution to* (4.1.6).

Proof. Let $\bar{x}$ be a cluster point of $\{x^k\}$, and let $S \subset \mathbb{N}$ be a subsequence such that $x^k \xrightarrow{S} \bar{x}$. By (4.3.5), $\{p^k\}$ is bounded so we may assume with no loss in generality that there is a $\bar{p}$ such that $p^k \xrightarrow{S} \bar{p}$. Theorem 4.3.3 tells us that $A\bar{x} + b = \bar{p}$ and $\hat{J}(\bar{x}, \bar{p}) = J_0(x^*)$ so that $J_0(\bar{x}) = \hat{J}(\bar{x}, A\bar{x} + b) = \hat{J}(\bar{x}, \bar{p}) = J_0(x^*)$. □

We now address the question of when the sequence $\{x^k\}$ has cluster points. For the IRWA of the previous section this question was answered by appealing to Theorem 4.2.7 which provided necessary and sufficient conditions for the compactness of the lower level sets of the function $J(x, \epsilon)$. This approach also applies to the ADAL algorithm, but it is heavy handed in conjunction with Assumption 4.3.2. In the next result we consider two alternative approaches to this issue.

Proposition 4.3.5. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. If either*

(a) $\left[\bar{x} \in X^\infty \cap \ker(H) \cap A^{-1}C^\infty \text{ satisfies } g^T\bar{x} \le 0\right] \iff \bar{x} = 0$, *or*

(b) Assumption 4.3.2 holds and

$$\left[\tilde{x} \in X^\infty \cap \ker(H) \cap \ker(A) \text{ satisfies } g^T\tilde{x} \le 0\right] \iff \tilde{x} = 0, \tag{4.3.17}$$

then $\{x^k\}$ is bounded and every cluster point of this sequence is a solution to (4.1.6).

Proof. Let us first assume that (a) holds. By Theorem 4.2.7, the condition in (a) (recall (4.2.13)) implies that the set $L(J(x^0, 0), 0)$ is compact. Hence, a solution x^* to (4.1.6) exists. By [73, Theorem 23.7], there exist p^* and u^* such that (x^*, p^*, u^*) satisfies (4.3.3), i.e., Assumption 4.3.2 holds. Since

$$\begin{aligned} J(x^k, 0) &= \varphi(x^k) + \operatorname{dist}\left(Ax^k + b \mid C\right) \\ &= \varphi(x^k) + \left\|(Ax^k + b) - P_C(Ax^k + b)\right\| \\ &\le \varphi(x^k) + \left\|(Ax^k + b) - p^k\right\| + \left\|p^k - P_C(p^k)\right\| + \left\|P_C(p^k) - P_C(Ax^k + b)\right\| \\ &= \hat{J}(x^k, p^k) + 2\left\|z^k\right\|, \end{aligned}$$

the second inequality in (4.3.8) tells us that for all $k \in \mathbb{N}$ we hvae

$$J(x^{k+1}, 0) \le \hat{J}(x^*, p^*) + 2\left\|z^k\right\| - (u^{k+1})^T z^{k+1} + \frac{1}{\mu}(q^k)^T(p^* - p^{k+1})).$$

By Lemma 4.3.1 and Theorem 4.3.3, the right-hand side of this inequality is bounded for all $k \in \mathbb{N}$, and so, by Theorem 4.2.7, the sequence $\{x^k\}$ is bounded. Corollary 4.3.4 then tells us that all cluster points of this sequence are solutions to (4.1.6).

Now assume that (b) holds. If the sequence $\{x^k\}$ is unbounded, then there is a subsequence $S \subset \mathbb{N}$ and a vector $\bar{x} \in X^\infty$ such that $\left\|x^k\right\|_2 \overset{S}{\to} \infty$ and $x^k / \left\|x^k\right\|_2 \overset{S}{\to} \bar{x}$ with $\|\bar{x}\|_2 = 1$. By Lemma 4.3.1, $\{p^k\}$ is bounded and, by Theorem 4.3.3, $z^k \to 0$. Hence, $(Ax^k + b - p^k)/\left\|x^k\right\|_2 = z^k/\left\|x^k\right\|_2 \overset{S}{\to} 0$ so that $A\bar{x} = 0$. In addition, the sequence $\{\hat{J}(x^k, p^k)\}$ is bounded, which implies $\hat{J}(x^k, p^k)/\left\|x^k\right\|_2^2 \overset{S}{\to} 0$ so that $H\bar{x} = 0$. Moreover, since H is positive semi-definite, $g^T(x^k/\left\|x^k\right\|_2) \le \hat{J}(x^k, p^k)/\left\|x^k\right\|_2 \overset{S}{\to} 0$ so that $g^T\bar{x} \le 0$. But then (b) implies that $\bar{x} = 0$. This contradiction implies that the sequence $\{x^k\}$ must be bounded. The result now follows from Corollary 4.3.4. $\square$

Note that, since $\ker(A) \subset A^{-1}C^\infty$, the condition given in (a) implies (4.3.17), and that (4.3.17) is strictly weaker whenever $\ker(A)$ is strictly contained in $A^{-1}C^\infty$.

We conclude this section by stating a result for the case when H is positive definite.

As has been observed, in such cases, the function J_0 is strongly convex and so the problem (4.1.6) has a unique global solution x^*. Hence, a proof paralleling that provided for Theorem 4.2.11 applies to give the following result.

Theorem 4.3.6. *Suppose that H is positive definite and the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, the problem* (4.1.6) *has a unique global solution x^* and $x^k \to x^*$.*

4.3.3 Complexity of ADAL

In this subsection, we analyze the complexity of ADAL. As was done for IRWA in Theorem 4.2.12, we show that ADAL requires at most $O(1/\varepsilon^2)$ iterations to obtain an ε-optimal solution to the problem (4.1.6). In contrast to this result, some authors [41, 42] establish an $O(1/\varepsilon)$ complexity for ε-optimality for ADAL-type algorithms applied to more general classes of problems, which includes (4.1.6). However, the ADAL decomposition employed by these papers involves subproblems that are as difficult as our problem (4.1.6), thereby rendering these decomposition unusable for our purposes. On the other hand, under mild assumptions, the recent results in [81] show that for a general class of problems, which includes (4.3.1), the ADAL algorithm employed here has $\hat{J}(x^k, p^k)$ converging to an ε-optimal solution to (4.3.1) with $O(1/\varepsilon)$ complexity in an *ergodic sense* and $\|Ax + b - p\|_2^2$ converging to a value less than ε with $O(1/\varepsilon)$ complexity. This corresponds to an $O(1/\varepsilon^2)$ complexity for ε-optimality for problem (4.1.6). As of this writing, we know of no result that applies to our ADAL algorithm that establishes a better iteration complexity bound for obtaining an ε-optimal solution to (4.1.6).

We use results in [81] to establish the following result.

Theorem 4.3.7. *Consider the problem* (4.1.6) *with $X = \mathbb{R}^n$ and suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$. Then, under Assumption 4.3.2, in at most $O(1/\varepsilon^2)$ iterations we have an iterate $x^{\bar{k}}$ with $k \leq \bar{k} \leq 2k - 1$ that is ε-optimal to* (4.1.6)*, i.e., such that* (4.2.17) *holds with $\tilde{x} = x^{\bar{k}}$.*

The key results from [81] used to prove this theorem follow.

Lemma 4.3.4. *[81, Lemma 2] Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$, and, under Assumption 4.3.2, let (x^*, p^*, u^*) be the optimal solution of (4.3.1). Then, for all $k \in \mathbb{N}$, we have*

$$\begin{aligned}\hat{J}(x^{k+1}, p^{k+1}) - \hat{J}(x^*, p^*) \leq{}& \tfrac{\mu}{2}(\left\|u^k\right\|_2^2 - \left\|u^{k+1}\right\|_2^2) - \tfrac{1}{2\mu}\left\|Ax^k + b - p^{k+1}\right\|_2^2 \\ &+ \tfrac{1}{2\mu}(\left\|Ax^* - Ax^k\right\|_2^2 - \left\|Ax^* - Ax^{k+1}\right\|_2^2).\end{aligned}$$

Lemma 4.3.5. *[81, Theorem 2] Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$, and, under Assumption 4.3.2, let (x^*, p^*, u^*) be the optimal solution of (4.3.1). Then, for all $k \in \mathbb{N}$, we have*

$$\left\|Ax^k + b - p^k\right\|_2^2 + \left\|Ax^k - Ax^{k-1}\right\|_2^2 \leq \tfrac{1}{k}[\left\|A(x^0 - x^*)\right\|_2^2 + \mu^2\left\|u^0 - u^*\right\|_2^2],$$

i.e., in particular, we have

$$\left\|Ax^k + b - p^k\right\|_2^2 \leq \tfrac{1}{k}[\left\|A(x^0 - x^*)\right\|_2^2 + \mu^2\left\|u^0 - u^*\right\|_2^2].$$

Remark 4.3.8. To see how the previous two lemmas follow from the stated results in [81], the table below provides a guide for translating between our notation and that of [81], which considers the problem

$$\min_{x,z}\ f(x) + g(z) \text{ subject to } Ax + Bz = c. \tag{4.3.18}$$

Problem (4.3.1)	Problem (4.3.18)
(x, p)	(z, x)
φ	g
$\operatorname{dist}(\cdot \mid C)$	f
A	B
$-I$	A
$-b$	c

For the results corresponding to our Lemmas 4.3.4 and 4.3.5, [81] requires f and g in (4.3.18) to be closed, proper, and convex functions. In our case, the corresponding functions $\operatorname{dist}(\cdot \mid C)$ and φ satisfy these assumptions.

By Lemma 4.3.3, the sequence $\{\omega^k\}$ is monotonically decreasing, meaning that $\{\|Ax^k - Ax^*\|_2^2\}$ and $\{\|u^k\|_2^2\}$ are bounded by some $\tau_1 > 0$ and $\tau_2 > 0$, respectively. The proof of Theorem 4.3.7 now follows as a consequence of the following lemma.

Lemma 4.3.6. *Suppose that the sequence $\{(x^k, p^k, u^k)\}$ is generated by ADAL with initial point $x^0 \in X$, and, under Assumption 4.3.2, let (x^*, p^*, u^*) be the optimal solution of (4.3.1). Moreover, let $\bar{k} \in K := \{k, k+1, \ldots, 2k-1\}$ be such that $\hat{J}(x^{\bar{k}}, p^{\bar{k}}) = \min_{k \in K} \hat{J}(x^k, p^k)$. Then,*

$$J_0(x^{\bar{k}}) - J_0(x^*) \leq \sqrt{\frac{l(\|A(x^0 - x^*)\|_2^2 + \mu^2 \|u^0 - u^*\|_2^2)}{k}} + \frac{\mu\tau_2 + \tau_1/\mu}{k}.$$

Proof. Summing the inequality in Lemma 4.3.4 for $j = k-1, \ldots, 2(k-1)$ yields

$$\begin{aligned}
&\left(\sum_{j=k-1}^{2k-2} \hat{J}(x^{j+1}, p^{j+1})\right) - k\hat{J}(x^*, p^*) \\
&\leq \tfrac{\mu}{2}\left(\left\|u^{k-1}\right\|_2^2 - \left\|u^{2k-1}\right\|_2^2\right) + \tfrac{1}{2\mu}\left(\left\|Ax^* - Ax^{k-1}\right\|_2^2 - \left\|Ax^* - Ax^{2k-1}\right\|_2^2\right) \\
&\leq \mu\tau_2 + \tau_1/\mu. \qquad (4.3.19)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\hat{J}(x^{\bar{k}}, p^{\bar{k}}) - \hat{J}(x^*, p^*) &= \min_{k \le j \le 2k-1} \hat{J}(x^j, p^j) - \hat{J}(x^*, p^*) \\
&\le \tfrac{1}{k} \sum_{j=k-1}^{2k-2} \hat{J}(x^{j+1}, p^{j+1}) - \hat{J}(x^*, p^*) \\
&\le \tfrac{1}{k}(\mu\tau_2 + \tau_1/\mu), \qquad (4.3.20)
\end{aligned}$$

where the last inequality follows from (4.3.19).

Next, observe that for any $x \in \mathbb{R}^n$ and p, we have

$$\begin{aligned}
J_0(x) - \hat{J}(x, p) &= \varphi(x) + \operatorname{dist}(Ax + b \mid C) - (\varphi(x) + \operatorname{dist}(p \mid C)) \\
&= \operatorname{dist}(Ax + b \mid C) - \operatorname{dist}(p \mid C) \\
&\le \|Ax + b - p\| \\
&= \sum_{i \in \mathcal{I}} \|A_i x + b_i - p_i\|_2 \\
&\le \sqrt{l}\, \|Ax + b - p\|_2\,, \qquad (4.3.21)
\end{aligned}$$

where the first inequality follows since $|\operatorname{dist}(z \mid C) - \operatorname{dist}(w \mid C)| \le \|z - w\|$, and the second follows by Jensen's inequality. Combining (4.3.20) and (4.3.21) gives

$$\begin{aligned}
J_0(x^{\bar{k}}) - J_0(x^*) &= J_0(x^{\bar{k}}) - \hat{J}(x^*, p^*) \\
&= J_0(x^{\bar{k}}) - \hat{J}(x^{\bar{k}}, p^{\bar{k}}) + \hat{J}(x^{\bar{k}}, p^{\bar{k}}) - \hat{J}(x^*, p^*) \\
&\le \sqrt{l} \left\| Ax^{\bar{k}} + b - p^{\bar{k}} \right\|_2 + \frac{\mu\tau_2 + \tau_1/\mu}{k} \\
&\le \sqrt{\frac{l(\|A(x^0 - x^*)\|_2^2 + \mu^2 \|u^0 - u^*\|_2^2)}{k}} + \frac{\mu\tau_2 + \tau_1/\mu}{k},
\end{aligned}$$

where the second inequality follows by Lemma 4.3.5 and the fact that $\bar{k} \ge k$. □

4.4 Nesterov Acceleration

In order to improve the performance of both IRWA and ADAL, one can use an acceleration technique due to Nesterov [61]. For the ADAL algorithm, we have implemented the acceleration as described in [43], and for the IRWA algorithm the details are given below. We conjecture that each accelerated algorithm requires $O(1/\varepsilon)$ iterations to produce an ε-optimal solution to (4.1.6), but this remains an open issue.

IRWA with Nesterov Acceleration

Step 0: (Initialization) Choose an initial point $x^0 \in X$, an initial relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$, and scaling parameters $\eta \in (0,1)$, $\gamma > 0$, and $M > 0$. Let $\sigma \geq 0$ and $\sigma' \geq 0$ be two scalars which serve as termination tolerances for the stepsize and relaxation parameter, respectively. Set $k := 0$, $y^0 := x^0$, and $t_1 := 1$.

Step 1: (Solve the re-weighted subproblem for x^{k+1})

Compute a solution x^{k+1} to the problem

$$\mathcal{G}(y^k, \epsilon^k): \quad \min_{x \in X} \hat{G}_{(y^k, \epsilon^k)}(x).$$

Let

$$t_{k+1} := \tfrac{1+\sqrt{1+4(t^k)^2}}{2}$$
$$\text{and} \quad y^{k+1} := x^{k+1} + \tfrac{t^k - 1}{t_{k+1}}(x^{k+1} - x^k).$$

Step 2: (Set the new relaxation vector ϵ^{k+1})

Set

$$\tilde{q}_i^k := A_i(x^{k+1} - y^k) \quad \text{and} \quad \tilde{r}_i^k := (I - P_{C_i})(A_i y^k + b_i) \quad \forall\, i \in \mathcal{I}_0.$$

If

$$\left\|\tilde{q}_i^k\right\|_2 \leq M\left[\left\|\tilde{r}_i^k\right\|_2^2 + (\epsilon_i^k)^2\right]^{\frac{1}{2}+\gamma} \quad \forall\, i \in \mathcal{I},$$

then choose $\epsilon^{k+1} \in (0,\ \eta\epsilon^k]$; else, set $\epsilon^{k+1} := \epsilon^k$. If $J(y^{k+1}, \epsilon^{k+1}) > J(x^{k+1}, \epsilon^{k+1})$, then set $y^{k+1} := x^{k+1}$.

Step 3: (Check stopping criteria)

If $\left\|x^{k+1} - x^k\right\|_2 \leq \sigma$ and $\left\|\epsilon^k\right\|_2 \leq \sigma'$, then stop; else, set $k := k+1$ and go to Step 1.

In this algorithm, the intermediate variable sequence $\{y^k\}$ is included. If y^{k+1} yields an objective function value worse than x^{k+1}, then we re-set $y^{k+1} := x^{k+1}$. This modification preserves the global convergence properties of the original version since

$$\begin{aligned}
& J(x^{k+1}, \epsilon^{k+1}) - J(x^k, \epsilon^k) \\
= & \ J(x^{k+1}, \epsilon^{k+1}) - J(y^k, \epsilon^k) + J(y^k, \epsilon^k) - J(x^k, \epsilon^k) \\
\leq & \ J(x^{k+1}, \epsilon^k) - J(y^k, \epsilon^k) \\
\leq & \ -\tfrac{1}{2}(x^{k+1} - y^k)^T \tilde{A}^T W_k \tilde{A}(x^{k+1} - y^k) \qquad (4.4.1) \\
= & \ -\tfrac{1}{2}(\tilde{q}^k)^T W_k \tilde{q}^k,
\end{aligned}$$

where the inequality (4.4.1) follows from Lemma 4.2.1. Hence, $\frac{1}{2}(\tilde{q}^k)^T W_k \tilde{q}^k$ is summable, as was required for Lemma 4.2.2 and Theorem 4.2.10.

4.5 Application to Systems of Equations and Inequalities

In this section, we discuss how to apply the general results from §4.2 and §4.3 to the particular case when H is positive definite and the system $Ax + b \in C$ corresponds a system of equations and inequalities. Specifically, we take $l = m$, $X = \mathbb{R}^n$, $C_i = \{0\}$ for $i \in \{1, \ldots, s\}$, and $C_i = \mathbb{R}_-$ for $i \in \{s+1, \ldots, m\}$ so that $C := \{0\}^s \times \mathbb{R}_-^{m-s}$ and

$$\begin{aligned}
J_0(x) &= \varphi(x) + \operatorname{dist}_1(Ax + b \mid C) \\
&= \varphi(x) + \sum_{i=1}^{s} |A_i x + b_i| + \sum_{i=s+1}^{m} (A_i x + b_i)_+. \qquad (4.5.1)
\end{aligned}$$

The numerical performance of both IRWA and ADAL on problems of this type will be compared in the following section. For each algorithm, we examine performance relative to a stopping criteria, based on percent reduction in the initial duality gap. It is straightforward to show that, since H is positive definite, the Fenchel-Rockafellar dual [73, Theorem 31.2] to (4.1.6) is

$$\begin{array}{ll} \underset{u}{\text{minimize}} & \frac{1}{2}(g + A^T u)^T H^{-1}(g + A^T u) - b^T u + \sum_{i\in\mathcal{I}} \delta^*\left(u_i \mid C_i\right) \\ \text{subject to} & u_i \in \mathbb{B}_2 \ \forall\, i \in \mathcal{I}, \end{array} \tag{4.5.2}$$

which in the case of (4.5.1) reduces to

$$\begin{array}{ll} \underset{u}{\text{minimize}} & \frac{1}{2}(g + A^T u)^T H^{-1}(g + A^T u) - b^T u \\ \text{subject to} & -1 \le u_i \le 1,\ i = 1, \ldots, s \\ & \quad 0 \le u_i \le 1,\ i = s+1, \ldots, m. \end{array}$$

In the case of linear systems of equations and inequalities, IRWA can be modified to improve the numerical stability of the algorithm. Observe that if both of the sequences $|r_i^k|$ and ϵ_i^k are driven to zero, then the corresponding weight w_i^k diverges to $+\infty$, which may slow convergence by unnecessarily introducing numerical instability. Hence, we propose a modification that addresses those iterations and indices $i \in \{s+1, \ldots, m\}$ for which $(A_i x^k + b_i)_- < 0$, i.e., those inequality constraint indices corresponding inequality constraints that are strictly satisfied (inactive). For such indices, it is not necessary to set $\epsilon_i^{k+1} < \epsilon_i^k$. There are many possible approaches to address this issue, one of which is given in the algorithm given below.

IRWA for Systems of Equations and Inequalities

Step 0: (Initialization) Choose an initial point $x^0 \in X$, initial relaxation vectors $\hat{\epsilon}^0 = \epsilon^0 \in \mathbb{R}^l_{++}$, and scaling parameters $\eta \in (0, 1)$, $\gamma > 0$, and $M > 0$. Let $\sigma \ge 0$ and $\sigma' \ge 0$ be two scalars which serve as termination tolerances for the stepsize and relaxation parameter, respectively. Set $k := 0$.

Step 1: (Solve the re-weighted subproblem for x^{k+1})

Compute a solution x^{k+1} to the problem

$$\mathcal{G}(x^k, \epsilon^k): \quad \min_{x \in X} \hat{G}_{(x^k,\epsilon^k)}(x).$$

Step 2: (Set the new relaxation vector ϵ^{k+1})

Set

$$q_i^k := A_i(x^{k+1} - x^k) \quad \text{and} \quad r_i^k := (I - P_{C_i})(A_i x^k + b_i) \quad \forall\, i = 0, \ldots, m.$$

If

$$\left\| q_i^k \right\|_2 \le M \left[\left\| r_i^k \right\|_2^2 + (\epsilon_i^k)^2 \right]^{\frac{1}{2}+\gamma} \quad \forall\, i = 1, \ldots, m, \tag{4.5.3}$$

then choose $\hat{\epsilon}^{k+1} \in (0,\ \eta\hat{\epsilon}^k]$ and, for $i = 1, \ldots, m$, set

$$\epsilon_i^{k+1} := \begin{cases} \hat{\epsilon}_i^{k+1} & , i = 1, \ldots, s, \\ \epsilon_i^k & , i > s \text{ and } (A_i x^k + b_i)_- \le -\hat{\epsilon}_i^k, \\ \hat{\epsilon}_i^{k+1} & , \text{otherwise.} \end{cases}$$

Otherwise, if (4.5.3) is not satisfied, then set $\hat{\epsilon}^{k+1} := \hat{\epsilon}^k$ and $\epsilon^{k+1} := \epsilon^k$.

Step 3: (Check stopping criteria)

If $\left\| x^{k+1} - x^k \right\|_2 \le \sigma$ and $\left\| \hat{\epsilon}^k \right\|_2 \le \sigma'$, then stop; else, set $k := k+1$ and go to Step 1.

Remark 4.5.1. In Step 2 of the algorithm above, the updating scheme for ϵ can be modified in a variety of ways. For example, one can also take $\epsilon_i^{k+1} := \epsilon_i^k$ when $i > s$ and $(A_i x^k + b_i)_- < 0$.

This algorithm yields the following version of Lemma 4.2.2.

Lemma 4.5.1. *Suppose that the sequence $\{(x^k, \epsilon^k)\}$ is generated by IRWA for Systems of Equations and Inequalities with initial point $x^0 \in X$ and relaxation vector $\epsilon^0 \in \mathbb{R}^l_{++}$, and,* *for $k \in \mathbb{N}$, let q_i^k and r_i^k for $i \in \mathcal{I}_0$ be as defined in Step 2 of the algorithm with*

$$q^k := ((q_0^k)^T, \ldots, (q_l^k)^T)^T \quad \textit{and} \quad r^k := ((r_0^k)^T, \ldots, (r_l^k)^T)^T.$$

Moreover, for $k \in \mathbb{N}$, define

$$w_i^k := w_i(x^k, \epsilon^k) \textit{ for } i \in \mathcal{I}_0 \quad \textit{and} \quad W_k := W(x^k, \epsilon^k),$$

and set $S := \left\{k \mid \epsilon^{k+1} \leq \eta\epsilon^k\right\}$. Then, the sequence $\{J(x^k, \epsilon^k)\}$ is monotonically decreasing. Moreover, either $\inf_{k\in\mathbb{N}} J(x^k, \epsilon^k) = -\infty$, in which case $\inf_{x\in X} J_0(x) = -\infty$, or the following hold:

1. $\sum_{k=0}^{\infty}(q^k)^T W_k q^k < \infty$.
2. $\hat{\epsilon}^k \to 0$ *and* $H(x^{k+1} - x^k) \to 0$.
3. $W_k q^k \xrightarrow{S} 0$.
4. $w_i^k r_i^k = r_i^k / \sqrt{\|r_i^k\|_2^2 + \epsilon_i^k} \in \mathbb{B}_2 \cap N\left(P_{C_i}(A_i x^k + b_i) \,|\, C_i\right)$, $i \in \mathcal{I}$, $k \in \mathbb{N}$.
5. $-\tilde{A}^T W_k q^k \in (\nabla\varphi(x^k) + \sum_{i\in\mathcal{I}} A_i^T w_i^k r_i^k) + N\left(x^{k+1} \,|\, X\right)$, $k \in \mathbb{N}$.
6. *If* $\{\text{dist}\left(Ax^k + b \mid C\right)\}_{k\in S}$ *is bounded, then* $q^k \xrightarrow{S} 0$.

Proof. Note that Lemma 4.2.1 still applies since it is only concerned with properties of the functions $\hat{G}$ and J. In addition, note that

$$\hat{\epsilon}^{k+1} \leq \hat{\epsilon}^k \quad \text{and} \quad \hat{\epsilon}^{k+1} \leq \epsilon^{k+1} \leq \epsilon^k \quad \forall\, k \geq 1.$$

With these observations, the proof of this lemma follows in precisely the same way as that of Lemma 4.2.2, except that in Part (2) $\{\hat{\epsilon}^k\}$ replaces $\{\epsilon^k\}$. □

With Lemma 4.5.1, it is straightforward to show that the convergence properties described in Theorems 4.2.10 and 4.2.11 also hold for the version of IRWA in this section.

4.6 Numerical Comparison of IRWA and ADAL

In this section, we compare the performance of our IRWA and ADAL algorithms in a set of three numerical experiments. The first two experiments involves cases where H is positive definite and the desired inclusion $Ax + b \in C$ corresponds to a system of equations and inequalities. Hence, for these experiments, we employ the version of IRWA as described for such systems in the previous section. In the first experiment, we fix the problem dimensions and compare the behavior of the two algorithms over 500 randomly generated problems. In the second experiment, we investigate how the methods behave when we scale up the problem size. For this purpose, we compare performance over 20 randomly generate problems of increasing dimension. The algorithms were implemented in Python using the NumPy and SciPy packages; in particular, we used the versions Python 2.7, Numpy 1.6.1, SciPy 0.12.0 [54, 64]. In both experiments, we examine performance relative to a stopping criteria based on percent reduction in the initial duality gap. In IRWA, the variables $\tilde{u}^k := W_k r^k$ are always dual feasible, i.e.,

$$\tilde{u}_i \in \mathbb{B}_2 \cap \operatorname{dom}\left(\delta^*\left(\cdot \mid C_i\right)\right) \quad \forall\, i \in \mathcal{I}$$

(recall Lemma 4.2.2(4)), and these variables constitute our kth estimate to the dual solution. On the other hand, in ADAL, the variables $\hat{u}^k = u^k - \frac{1}{\mu} q^k$ are always dual feasible (recall Lemma 4.3.1), so these constitute our kth estimate to the dual solution for this algorithm. The duality gap at any iteration is the sum of the primal and dual objectives at the current primal-dual iterates.

In both IRWA and ADAL, we solve the subproblems using CG, which is terminated when the ℓ_2-norm of the residual is less than 10% of the norm of the initial residual. At each iteration, the CG algorithm is initiated at the previous step x^{k-1}. In both experiments, we set $x^0 := 0$, and in ADAL we set $u^0 := 0$. It is worthwhile to note that we have observed that the performance of IRWA is sensitive to the initial choice of ϵ^0 while ADAL is sensitive to μ. We do not investigate this sensitivity in detail when presenting the results of our

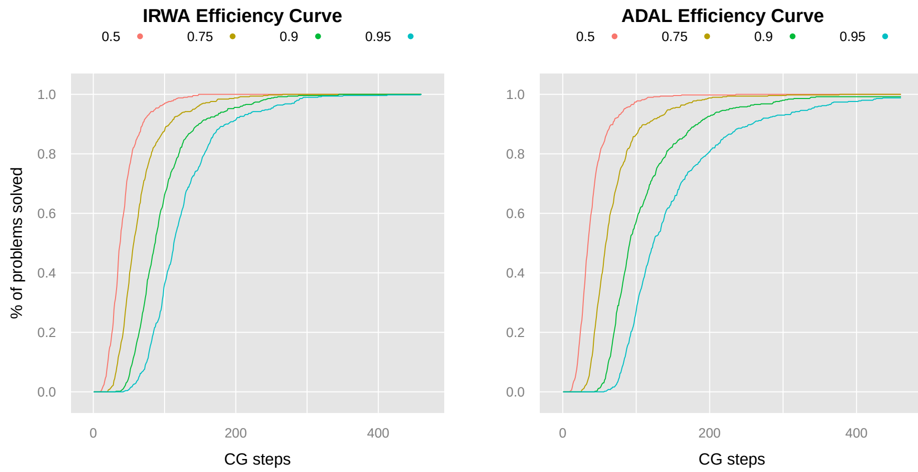


Figure 4.1: Efficiency curves.

experiments, and we have no theoretical justification for our choices of these parameters. However, we empirically observe that these values should increase with dimension. For each method, we have chosen an automatic procedure for initializing these values that yields good overall performance. The details are given in the experimental descriptions. More principled methods for initializing and updating these parameters is the subject of future research.

In the third experiment, we apply both algorithms to an l_1 support vector machine (SVM) problem. Details are given in the experimental description. In this case, we use the stopping criteria as stated along with the algorithm descriptions in the chapter, i.e., not criteria based on a percent reduction in duality gap. In this experiment, the subproblems are solved as in the first two experiments with the same termination and warm-start rules.

First Experiment: In this experiment, we randomly generated 500 instances of problem (4.5.1). For each, we generated $A \in \mathbb{R}^{600\times1000}$ and chose C so that the inclusion $Ax+b \in C$ corresponded to 300 equations and 300 inequalities. Each matrix A is obtained by first randomly choosing a mean and variance from the integers on the interval $[1, 10]$ with equal probability. Then the elements of A are chosen from a normal distribution having this mean and variance. Similarly, each of the vectors b and g are constructed by first randomly

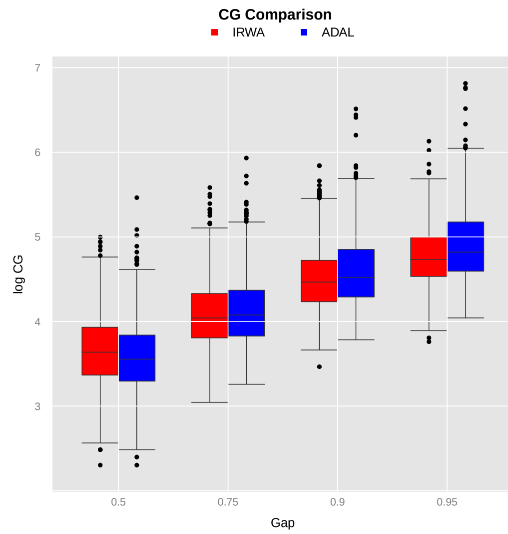


Figure 4.2: Box plot of CG steps for each duality gap threshold.

choosing integers on the intervals $[-100, 100]$ for the mean and $[1, 100]$ for the variance with equal probability and then obtaining the elements of these vectors from a normal distribution having this mean and variance. Each matrix H had the form $H = 0.1I + LL^T$ where the elements of $L \in \mathbb{R}^{n\times n}$ are chosen from a normal distribution having mean 1 and variance 2. For the input parameters for the algorithms, we chose $\eta := 0.6$, $M := 10^4$, $\gamma := \frac{1}{6}$, $\mu := 100$, and $\epsilon_i^0 := 2000$ for each $i \in \mathcal{I}$. Efficiency curves for both algorithms are given in Figure 4.1, which illustrates the percentage of problems solved verses the total number of CG steps required to reduce the duality gap by 50, 75, 90 and 95 percent. The percentage of the 500 problems solved is plotted verses the total number of CG steps. IRWA terminated in fewer than 460 CG steps on all problems. ADAL required over 460 CG steps on 8 of the problems. The greatest number of CG steps required by IRWA was 460 when reducing the duality gap by 95%. ADAL stumbled at the 95% level on 8 problems, requiring $609, 494, 628, 674, 866, 467, 563, 856, 676$ and 911 CG steps for these problems. Figure 4.2 contains a box plot for the log of the number of CG iterations required by each algorithm for each of the selected accuracy levels. Overall, in this experiment, the methods seem comparable with a slight advantage to IRWA in both the mean and variance of the

Figure 4.3: CG steps for each duality gap and CPU time plot as dimension increases.

number of required CG steps.

Second Experiment: In the second experiment, we randomly generated 20 problems of increasing dimention. The numbers of variables were chosen to be $n = 200+500(j-1),\ j = 1, \ldots, 20$, where for each we set $m := n/2$ so that the inclusion $Ax + b \in C$ corresponds to equal numbers of equations and inequalities. The matrix A was generated as in the first experiment. Each of the vectors b and g were constructed by first choosing integers on the intervals $[-200, 200]$ for the mean and $[1, 200]$ for the variance with equal probability and then obtaining the elements of these vectors from a normal distribution having this mean and variance. Each matrix H had the form $H = 40I + LDL^T$, where $L \in \mathbb{R}^{n \times k}$ with $k = 8$ and D was diagonal. The elements of L were constructed in the same way as those of A, and those of D were obtained by sampling from the inverse gamma distribution $f(x) := \frac{b^a}{\Gamma(a)} x^{-a-1} e^{-b/x}$ with $a = 0.5,\ b = 1$. We set $\eta := 0.5$, $M := 10^4$, and $\gamma := \frac{1}{6}$, and for each $j = 1, \ldots, 20$ we set $\epsilon_i^0 := 10^{2+1.3\ln(j+10)}$ for each $i = 1, \ldots, m$, and $\mu := 500(1+j)$. In Figure 4.3, we present two plots showing the number of CG steps and the log of the CPU times versus variable dimensions for the two methods. The plots illustrate that the algorithms performed similarly in this experiment.

Third Experiment: In this experiment, we solve the l_1-SVM problem as introduced in

[51]. In particular, we consider the exact penalty form

$$\min_{\beta \in \mathbb{R}^n} \sum_{i=1}^{m} \left(1 - y_i \left(\sum_{j=1}^{n} x_{ij} \beta_j \right) \right)_+ + \lambda \|\beta\|_1 , \tag{4.6.1}$$

where $\{(\boldsymbol{x_i}, y_i)\}_{i=1}^m$ are the training data points with $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ for each $i = 1, \ldots, m$, and λ is the penalty parameter. In this experiment, we randomly generated 40 problems in the following way. First, we sampled an integer on $[1, 5]$ and another on $[6, 10]$, both from uniform distributions. These integers were taken as the mean and standard deviation of a normal distribution, respectively. We then generated an $m \times s$ component-wise normal random matrix T, where s was chosen to be $19+2j,\ j = 0, 1, \ldots, 39$ and m was chosen to be $200 + 10j,\ j = 0, 1, \ldots, 39$. We then generated an s-dimensional integer vector $\hat{\beta}$ whose components were sampled from the uniform distribution on the integers between -100 and 100. Then, y_i was chosen to be the sign of the i-th component of $T\hat{\beta}$. In addition, we generated an $m \times t$ i.i.d. standard normal random matrix R, where t was chosen to be $200 + 30j,\ j = 0, 1, \ldots, 39$. Then, we let $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^T := [T, R]$. For all 40 problems, we fixed the penalty parameter at $\lambda = 50$. In this application, the problems need to be solved exactly, i.e., a percent reduction in duality gap is insufficient. Hence, in this experiment, we use the stopping criteria as described in Step 3 of both IRWA and ADAL. For IRWA, we set $\epsilon_i^0 := 10^4$ for all $i \in \mathcal{I}$, $\eta := 0.7$, $M := 10^4$, $\gamma := \frac{1}{6}$, $\sigma := 10^{-4}$ and $\sigma' := 10^{-8}$. For ADAL, we set $\mu := 1$, $\sigma := 0.05$ and $\sigma'' := 0.05$. We also set the maximum iteration limit for ADAL to 150. Both algorithms were initialized at $\beta := 0$. Figure 4.4 has two plots showing the objective function values of both algorithms at termination, and the total CG steps taken by each algorithm. These two plots show superior performance for IRWA when solving these 40 problems.

Based on how the problems were generated, we would expect the non-zero coefficients of the optimal solution β to be among the first $s = 19 + 2j,\ j = 0, \ldots, 39$ components corresponding to the matrix T. To investigate this, we considered "zero" thresholds of $10^{-3}, 10^{-4}$ and 10^{-5}; i.e., we considered a component as being "equal" to zero if its absolute value was less than a given threshold. Figure 4.5 shows a summary of the number of unexpected zeros for each algorithm. For both thresholds 10^{-4} and 10^{-5}, IRWA yields fewer false positives in terms of the numbers of "zero" values computed. The numbers of false positives is similar for the threshold 10^{-3}. At the threshold 10^{-5}, the difference in recovery is dramatic with IRWA always having fewer than 14 false positives while ADAL has a median of about 1000 false positives. These plots show that IRWA has significantly fewer false positives for the nonzero components, and in this respect returned preferable sparse recovery results over ADAL in this experiment.

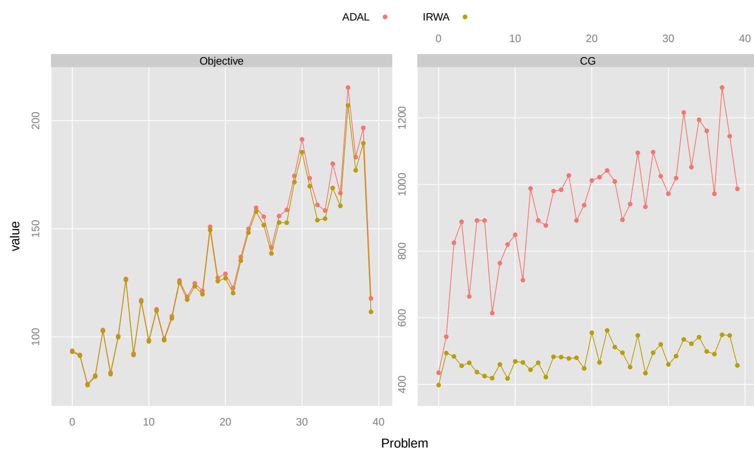


Figure 4.4: CG Steps and Objective function Comparison.

Finally, we use this experiment to demonstrate Nesterov's acceleration for IRWA. The effect on ADAL has already been shown in [43], so we only focus on the effect of accelerating IRWA. The 40 problems were solved using both IRWA and accelerated IRWA with the parameters stated above. Figure 4.6 shows the differences in objective function values (left panel) obtained by normal and accelerated IRWA ($\frac{\text{normal}-\text{accelerated}}{\text{accelerated}} \times 100$), and differences in numbers of CG steps (right panel) required to converge to the objective function value in the left panel (normal$-$accelerated). Accelerated IRWA always converged to a point with a smaller objective function value, and accelerated IRWA typically required fewer CG steps. (There was only one exception, the last problem, where accelerated IRWA required two more CG steps.) The graphs show that accelerated IRWA performs

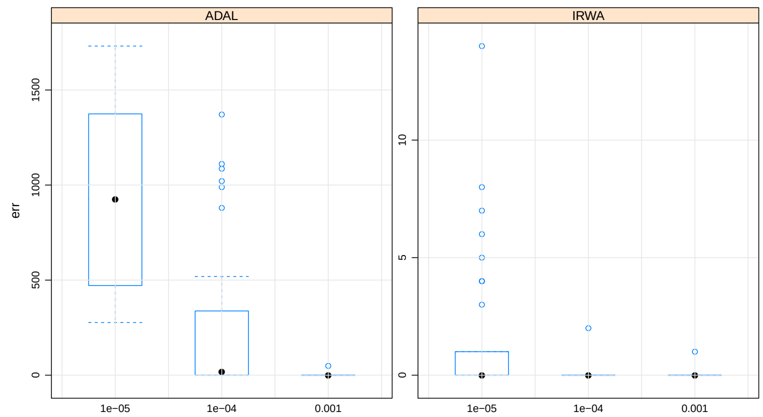


Figure 4.5: Sparsity of the Solutions

significantly better than unaccelerated IRWA in terms of both objective function values obtained and CG steps required.

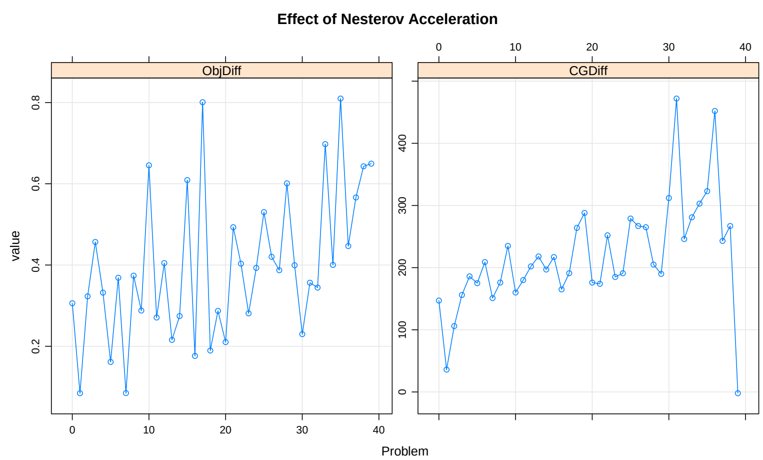


Figure 4.6: Performance of normal IRWA and Accelerated IRWA

Chapter 5

A Dynamic Penalty Parameter Updating Strategy for Matrix-Free SQO

5.1 Introduction

In this chapter, we consider the use of SQO methods for solving large-scale instances of nonlinear optimization problems (NLPs). While they have proved to be effective for solving small- to medium-scale problems, SQO methods have traditionally faltered in large-scale settings due to the expense of (accurately) solving large-scale quadratic subproblems (QPs) during each iteration. However, with the use of matrix-free methods for solving the subproblems, one may consider the acceptance of inexact subproblem solutions. Such a feature offers the possibility of terminating the subproblem solver early, perhaps well before an accurate solution has been computed. This characterizes the types of strategies that we propose in this chapter.

Recently, some work has been done to provide global convergence guarantees for SQO methods that allow inexact subproblem solves [27]. However, the practical efficiency of such an approach remains an open question. A critical aspect of any implementation of such an approach is the choice of subproblem solver. This is the case as the solver must be able to provide good inexact solutions quickly, as well as have the ability to compute highly accurate solutions—say, by exploiting well-chosen starting points—in the neighborhood of a solution of the NLP. In addition, while a global convergence mechanism such as a merit function or filter is necessary to guarantee convergence from remote starting points, an NLP algorithm can suffer when such a mechanism does not immediately guide the algorithm toward promising regions of the search space. To confront this issue when an exact penalty function is used as a merit function, we propose a dynamic penalty parameter updating strategy to be incorporated *within* the subproblem solver so that each computed search direction predicts progress toward both feasibility and optimality. This strategy represents a stark contrast to previously proposed techniques that only update the penalty parameter after a sequence of iterations [40] or at the expense of numerous subproblem solves within a single iteration [19, 13].

We organize this chapter as follows. In the remainder of this section, we outline our notation and introduce various definitions and concepts that will be employed throughout the chapter. In §5.2, we introduce a basic penalty-SQO algorithm that will form the framework for which we will introduce our penalty parameter updating strategy (see §5.3) and matrix-free subproblem solvers (see §5.4). We discuss implementations of our methods in §5.5 and the results of extensive numerical experiments in §5.6.

5.1.1 Notation

We briefly summarize the notation used in this chapter. Let $\mathbb{R}^n$ be the space of real n-vectors, $\mathbb{R}^n_+$ be the nonnegative orthant of $\mathbb{R}^n$ (i.e., $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geq 0\}$), and $\mathbb{R}^n_{++}$ be the interior of $\mathbb{R}^n_+$ (i.e., $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x > 0\}$). The set of $m \times n$ real matrices is denoted $\mathbb{R}^{m\times n}$. On $\mathbb{R}^n$, the ℓ_2 (i.e., Euclidean) norm is indicated as $\|\cdot\|_2$, with the unit ℓ_2-norm ball defined as $\mathbb{B}_2 := \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$. For a pair of vectors $\{u, v\} \subset \mathbb{R}^n$, their inner product is written as $\langle u, v\rangle := u^T v$ and the line segment between them is written as $[u, v]$. The middle value operator of $\{a, b, c\} \subset \mathbb{R}$, denoted by $\text{mid}\{a, b, c\}$, returns the median of

$\{a, b, c\}$. For a scalar a, let $(a)_+ = \max\{a, 0\}$ and $(a)_- = \min\{a, 0\}$. The set of natural numbers is denoted by $\mathbb{N}$.

For a set of scalars $b_i \in \mathbb{R}$ for $i \in \{1, \ldots, m\}$, we use bold lettering to denote the vector $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T \in \mathbb{R}^m$. (For convenience, we also use $\mathbf{1}_n$ to denote the n-vector of all ones and $\mathbf{0}_n$ to denote the n-vector of all zeros.) Similarly, given vectors $y_i \in \mathbb{R}^{d_i}$ for $i \in \{1, \ldots, m\}$, we often find it convenient to refer to the element $\mathbf{y} = (y_1, \ldots, y_m)$ on the product space $\mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}$. Conversely, given $\mathbf{y} \in \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}$, the i-th component of $\mathbf{y}$ (an element of $\mathbb{R}^{d_i}$) is denoted y_i while the j-th component of y_i is written as y_{ij}. In the product space, we define the norm

$$\|\mathbf{y}\| = \|(y_1, \ldots, y_m)\| := \sum_{i=1}^{m} \|y_i\|_2 \,,$$

of which the dual norm can be verified to be

$$\|\mathbf{y}\|_* = \sup_{i \in \{1, \ldots, m\}} \|y_i\|_2 \,.$$

For convex sets $C_i \in \mathbb{R}^{d_i}$ for $i \in \{1, \ldots, m\}$, we define the set

$$\mathbf{C} := C_1 \times \cdots \times C_m \subset \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}.$$

We define the distance functions

$$\mathrm{dist}_2\left(y_i \mid C_i\right) := \inf_{z_i \in C_i} \|y_i - z_i\|_2 \quad \text{and} \quad \mathrm{dist}\left(\mathbf{y} \mid \mathbf{C}\right) := \sum_{i=1}^{l} \mathrm{dist}_2\left(y_i \mid C_i\right)$$

as well as the corresponding projection operators

$$P_{C_i}(y_i) := \arg\min_{z_i \in C_i} \|z_i - y_i\|_2 \quad \text{and} \quad P_{\mathbf{C}}(\mathbf{y}) := \arg\min_{\boldsymbol{z} \in \mathbf{C}} \|\boldsymbol{z} - \mathbf{y}\|.$$

For an extended-real-valued function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, the Legendre-Fenchel conjugate of

f is denoted as $f^\star$. For a convex set $X \subset \mathbb{R}^n$, we define the indicator function

$$\delta(x|X) := \begin{cases} 0 & \text{if } x \in X, \\ \infty & \text{otherwise.} \end{cases}$$

The conjugate of $\delta(\cdot|X)$ is the support function of X, which we denote by

$$\delta^\star(x|X) = \sup_{x \in X} \langle y, x \rangle.$$

If f is convex, then the subdifferential of f at x is defined as the set

$$\partial f(x) := \{y \in \mathbb{R}^n : f(x) + \langle y, z - x \rangle \leq f(z) \text{ for all } z \in \mathbb{B}_2^n\}.$$

For example, the subdifferentials of our distance functions are given by (see [74])

$$\partial \text{dist}_2\left(y_i \mid C_i\right) := \begin{cases} \frac{(I - P_{C_i}) y_i}{\|(I - P_{C_i}) y_i\|_2} & \text{if } i \notin \mathcal{A}(\mathbf{y}) \\ \mathbb{B}_2 \cap N(y_i|C_i) & \text{if } i \in \mathcal{A}(\mathbf{y}), \end{cases}$$

$$\text{and} \quad \partial \text{dist}\left(\mathbf{y} \mid C\right) := \partial \text{dist}_2\left(y_1 \mid C_1\right) \times \cdots \times \partial \text{dist}_2\left(y_m \mid C_m\right),$$

where the index set $\mathcal{A}(\mathbf{y})$ is defined as

$$\mathcal{A}(\mathbf{y}) := \{i \in \{1, ..., m\} : \text{dist}_2\left(y_i \mid C_i\right) = 0\}$$

and the normal cone to C_i at y_i is defined by

$$N(y_i|C_i) := \{z_i \in \mathbb{R}^{d_i} : \langle z_i, p_i - y_i \rangle \leq 0 \text{ for all } p_i \in C_i\}.$$

5.2 A Penalty-SQO Framework

In this section, we formulate our problem of interest and outline the basic components of a penalty-SQO algorithm [34]. This method represents the framework in which we will define our dynamic penalty parameter updating strategy and matrix-free solvers whose purposes, respectively, are to guide the algorithm toward promising areas of the search space and approximately solve the arising direction-finding subproblems.

We formulate our problem of interest as the following nonlinear optimization problem with equality and inequality constraints for which we assume that the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t. } & c_i(x) = 0, \ i \in \{1, \dots, \bar{m}\} \\ & c_i(x) \leq 0, \ i \in \{\bar{m}+1, \dots, m\}. \end{aligned} \tag{NLP}$$

For our penalty-SQO framework, we define two functions for use in the algorithm and for characterizing first-order stationary solutions. First, with a penalty parameter $\rho \in \mathbb{R}_+$, we define measure of infeasibility and exact penalty function

$$v(x) = \sum_{i=1}^{\bar{m}} \|c_i(x)\|_2 + \sum_{i=\bar{m}+1}^{m} (c_i(x))_+ \quad \text{and} \quad \phi(x;\rho) = \rho f(x) + v(x).$$

Generally speaking, our penalty-SQO framework aims to solve (NLP) through systematic minimization of $\phi(x;\rho)$ for appropriately chosen values of $\rho > 0$. However, if the constraints of (NLP) are infeasible, then the algorithm is designed to return an infeasibility certificate in the form of a stationary point for $\phi(x;0) = v(x)$. Second, we define the Fritz John function for (NLP) which, given $\rho \in \mathbb{R}_+$ and $\eta \in \mathbb{R}^m$, is

$$F(x;\rho,\eta) = \rho f(x) + \langle \eta, c(x) \rangle.$$

We remark that we have defined $\rho \in \mathbb{R}_+$ as having the dual role of penalty parameter in ϕ and objective multiplier in F. In fact, this makes sense from both theoretical and practical perspectives. First-order stationarity conditions for (NLP) can be written in terms of ∇F, the constraint function c, and bounds on the dual variables; see [27].

In the k-th iteration of our penalty-SQO framework, the search direction computation is based on a local model of the penalty function about the current primal iterate $x^k \in \mathbb{R}^n$ that makes use of the dual iterate $\eta^k \in \mathbb{R}^m$. We define this model as

$$J(d; x^k, \rho, \eta^k) := \rho \nabla f(x^k)^T d + d^T H(x^k, \rho, \eta^k) d + \sum_{i=1}^{\bar{m}} |c_i(x^k) + \nabla c_i(x^k)^T d| + \sum_{i=\bar{m}+1}^{m} \max\{c_i(x^k) + \nabla c_i(x^k)^T d, 0\},$$

where H represents an approximation of $\nabla^2_{xx} F$ so that

$$H(x^k, \rho, \eta^k) \approx \nabla^2_{xx} F(x^k, \rho, \eta^k) = \rho \nabla^2_{xx} f(x^k) + \sum_{i=1}^{m} \eta_i^k \nabla^2_{xx} c_i(x^k).$$

Ultimately, the search direction computed in the k-th iteration represents, for some $X \subset \mathbb{R}^n$ containing $\{0\}$ and penalty parameter $\rho^k > 0$, an approximate solution of the subproblem

$$\min_{d \in X} \; J(d; x^k, \rho^k, \eta^k). \tag{QP}$$

(We introduce the set X to allow for the possibility of employing, say, a trust region constraint; e.g., for some $\Delta > 0$, we may define X such that $X \subset \{d : \|d\|_2 \leq \Delta\}$). In fact, the value $\rho^k \in (0, \rho^{k-1}]$ is to be computed *during* the subproblem solve in order to satisfy two critical inequalities. First, as is typical in the context of a penalty-SQO method, we intend for the pair (d^k, ρ^k) to be computed such that the search direction d^k is a direction of sufficient descent for $\phi(\cdot; \rho^k)$ from x^k. This is guaranteed if the reduction in the local model $J(\cdot; x^k, \rho^k, \eta^k)$ yielded by d^k, namely,

$$\Delta J(d; x^k, \rho^k, \eta^k) = J(0; x^k, \rho^k, \eta^k) - J(d^k; x^k, \rho^k, \eta^k),$$

is sufficiently positive. This requirement—which we formulate concretely in §5.3.2—represents the first critical inequality yielded by our search direction. In fact, the second critical inequality that we impose is similar, except that it relates to the reduction yielded by d^k in the local model $J(\cdot; x^k, 0, \eta^k)$ along d^k, namely,

$$\Delta J(d; x^k, 0, \eta^k) = J(0; x^k, 0, \eta^k) - J(d; x^k, 0, \eta^k).$$

Our dynamic strategy for setting ρ^k is designed so that this reduction is also sufficiently positive at the end of the search direction computation, implying that d^k is also a direction of sufficient descent from x^k for $\rho(\cdot; 0) = v$.

Overall, the k-th iteration of our penalty-SQO strategy proceeds in the following manner. First, a search direction and penalty parameter pair (d^k, ρ^k) is computed by a subproblem solver such that the resulting search direction yields reductions in the models of the penalty function and measure of infeasibility that are sufficiently large. Then—potentially after additional updates of the penalty parameter—a line search may be performed for the merit function $\phi(\cdot; \rho^k)$ from x^k along the search direction d^k, yielding the step-size $\alpha^k \in \mathbb{R}_{++}$. Finally, the new iterate is set as $x^{k+1} \leftarrow x^k + \alpha^k d^k$ and the algorithm proceeds to the $(k+1)$-st iteration. We state this framework in (2)

Algorithm 2 A Framework of *S*equential *Qu*adratic Optimization Algorithm

1. (Initialization) Choose $\gamma, \theta_l \in (0, 1)$. Set $k \leftarrow 0$ and choose (x^k, ρ^k, η^k).
2. (Subproblem Solution) Solve (QP) to obtain a primal-dual solution (d^k, η^k) and the penalty parameter ρ^k.
3. (Line Search) Let α^k be the largest value in $\{\gamma^0, \gamma^1, \gamma^2, \dots\}$ such that

$$\phi(x^k + \alpha^k d^k, \rho_{k+1}) - \phi(x^k, \rho_{k+1}) \leq -\theta_l \alpha^k \Delta J(d; x^k, \rho^k, \eta^k).$$

4. Set $x^{k+1} \leftarrow x^k + \alpha^k d^k$ and $k \leftarrow k + 1$ and go to step 1.

5.3 A Dynamic Penalty Parameter Updating Strategy

In this section, we present our new dynamic penalty parameter updating strategy. Our method is unique in that it is intended to be employed within a solver for the subproblem arising in our penalty-SQO framework. A potential pitfall of such an approach is that, since the penalty parameter dictates the weight between the objective terms in (QP), one may disrupt typical convergence guarantees of the subproblem solver by manipulating this weight during the solution process. However, under reasonable assumptions, we prove that for sufficiently small values of the penalty parameter, our updating strategy will no longer be triggered. Consequently, once the penalty parameter reaches a sufficiently small value, it will remain fixed and the subproblem solver will effectively be applied to solve (QP) for a fixed value of ρ^k. We state our proposed updating strategy in such a way that it can be incorporated into various subproblem solvers; see §5.4.

5.3.1 Preliminaries

As our penalty parameter updating strategy is to be employed in each iteration of our penalty-SQO framework, we can present our strategy generically by focusing on the k-th iteration of the framework. Thus, for ease of exposition in this section, we utilize the following shorthand notation to drop the dependence of certain quantities on the iteration number:

$$\begin{aligned} &g = \nabla f(x^k),\ a_i = \nabla c_i(x^k),\ b_i = c_i(x^k),\ A = [a_1, \cdots, a_m]^T, \\ &H_f \approx \nabla^2_{xx} f(x^k),\ H_0 \approx \sum_{i=1}^{m} \eta_i^k \nabla^2_{xx} c_i(x^k),\ H_\rho = \rho H_f + H_0. \end{aligned} \tag{5.3.1}$$

We also make the following assumption about the subproblem data.

Assumption 5.3.1. *The subproblem data matrices A, H_f, and H_0 are such that*

(i) H_ρ is positive definite for any $\rho \in [0, \rho^{k-1}]$; and

(ii) $\|a_i\|_2 > 0$ for all $i \in \{1, \ldots, m\}$.

We claim that this assumption is reasonable due to the following considerations. First, in large-scale contexts, it is typically impractical or inefficient to construct complete second-derivative matrices. Hence, as indicated in (5.3.1), we assume that H_f and H_0 represent approximate (low-rank) Hessian matrices with at least H_0 being positive definite. (See §5.5.3 for further discussion of such approximations.) Second, if $a_i = 0$ for any $i \in \{1, \ldots, m\}$, then the i-th constraint in the subproblem is superfluous and can be removed from consideration. Such a phenomenon can be detected during a preprocessing phase for the subproblem, so for simplicity in our discussion we assume that each constraint gradient is nonzero. As such, for notational convenience, we may define the scaled quantities $\bar{a}_i := a_i/\|a_i\|_2$ and $\bar{b}_i := b_i/\|a_i\|_2$ for all $i \in \{1, \ldots, m\}$.

Of central importance in the subproblems are the convex sets

$$\begin{aligned} C_i &:= \{d \in \mathbb{R}^n : a_i^T d + b_i = 0\}, \quad i \in \{1, \ldots, \bar{m}\} \\ \text{and} \quad C_i &:= \{d \in \mathbb{R}^n : a_i^T d + b_i \leq 0\}, \quad i \in \{\bar{m}+1, \ldots, m\}. \end{aligned}$$

The penalty term in the model J can thus be written as

$$\sum_{i=1}^{m} \|a_i\|_2 \operatorname{dist}_2(d \mid C_i),$$

meaning that, without loss of generality (i.e., assuming $\|a_i\|_2 = 1$ for all $i \in \{1, \ldots, m\}$) we may rewrite (in shorthand) the penalty-SQO subproblem (QP) as

$$\min_{d \in \mathbb{R}^n} J(d; \rho), \quad \text{where} \quad \begin{cases} J(d;\rho) = \varphi(d;\rho) + \displaystyle\sum_{i=1}^{m} \operatorname{dist}_2(d \mid C_i) + \delta(d|X) \\ \varphi(d;\rho) = \rho g^T d + \frac{1}{2} d^T H_\rho d. \end{cases} \tag{QP$_\rho$}$$

We refer to (QP$_\rho$) with $\rho > 0$ as a *penalty subproblem*, whereas we refer to (QP$_\rho$) with $\rho = 0$ as the *feasibility subproblem*. Direct calculation shows the Fenchel–Rockafellar dual of subproblem (QP$_\rho$) is given by

$$\max_{\mathbf{u} \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n} D(\mathbf{u};\rho) \quad \text{s.t.} \quad u_{m+1} = \sum_{i=0}^{m} u_i \quad \text{and} \quad u_i \in \mathbb{B}_2 \quad \text{for all} \quad i \in \{1, \ldots, m\}, \tag{DQP$_\rho$}$$

where the dual objective function is given by

$$D(\mathbf{u};\rho) = -\frac{1}{2}(u_0 - \rho g)^T H_\rho^{-1}(u_0 - \rho g) - \sum_{i=1}^{m} \delta^*(u_i|C_i) - \delta^*(u_{m+1}|X).$$

An interesting aspect of the dual subproblem (DQP$_\rho$) is that the penalty parameter appears only in the objective function; thus, if $\mathbf{u}$ satisfies the constraints of (DQP$_\rho$), then it is dual-feasible regardless of the value of ρ appearing in the subproblem. As a result, by weak duality, we have for any primal-dual feasible pair $(d, \mathbf{u})$ that both

$$D(\mathbf{u}; 0) \leq J(d; 0) \quad \text{and} \quad D(\mathbf{u}; \rho) \leq J(d; \rho). \tag{5.3.2}$$

We close this subsection by noting that projections onto the set C_i for any $i \in \{1, \ldots, m\}$ is especially easy to compute; in particular,

$$\begin{aligned} P_{C_i}(x) &= x - (\bar{a}_i^T x + \bar{b}_i)\bar{a}_i, \quad && i \in \{1, \ldots, \bar{m}\} \\ \text{and} \quad P_{C_i}(x) &= x - (\bar{a}_i^T x + \bar{b}_i)_+ \bar{a}_i, \quad && i \in \{\bar{m}+1, \ldots, m\}. \end{aligned}$$

We also observe that if $X = \mathbb{R}^n$, then the Fenchel-Rockafellar dual of (QP_ρ) reduces to

$$\max_{\mathbf{u} \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n} D(\mathbf{u}; \rho) \quad \text{s.t.} \quad 0 = \sum_{i=0}^{m} u_i \quad \text{and} \quad u_i \in \mathbb{B}_2 \quad \text{for all} \quad i \in \{1, \ldots, m\},$$

where

$$D(\mathbf{u}; \rho) = -\frac{1}{2}(u_0 - \rho g)^T H_\rho^{-1}(u_0 - \rho g) - \sum_{i=1}^{m} \delta^*(u_i | C_i).$$

5.3.2 Updating ρ

We are now prepared to present our dynamic penalty parameter updating strategy. For a given $\rho > 0$, let $(d_\rho^*, \mathbf{u}_\rho^*)$ represent an optimal primal-dual pair for the penalty subproblem corresponding to ρ; in particular, $(d_0^*, \mathbf{u}_0^*)$ represents an optimal primal-dual pair for the feasibility subproblem. We present our algorithm in the context of a subproblem solver that generates two sequences of iterates: The first sequence of iterates, call it $\{(d^{(j)}, \mathbf{u}^{(j)})\}$, represents a sequence of primal-dual feasible solution estimates for a penalty subproblem, while the second sequence of iterates, call it $\{\mathbf{w}^{(j)}\}$, represents a sequence of dual feasible solution estimates for the feasibility subproblem. (In our strategy, we do not make separate use of a sequence of primal solution estimates for the feasibility subproblem; rather, the sequence $\{d^{(j)}\}$ plays this role as well.) Without loss of generality, we assume that the j-th dual solution estimate $\mathbf{w}^{(j)}$ represents a better (or no worse) dual solution for the feasibility subproblem than $\mathbf{u}^{(j)}$ in the sense that $D(\mathbf{w}^{(j)};0) \geq D(\mathbf{u}^{(j)};0)$. This is a reasonable assumption since if this inequality were not to hold, then one could simply replace $\mathbf{w}^{(j)}$ with $\mathbf{u}^{(j)}$ as the j-th dual feasible solution estimate for the feasibility subproblem.

Observe that, by the definition of the model J, we have

$$J^{(0)} := J(0;0) = J(0;\rho) = \sum_{i=1}^{\bar{m}} |\bar{b}_i| + \sum_{i=\bar{m}+1}^{m} (\bar{b}_i)_+ \geq 0$$

for any $\rho > 0$. We then define, for any given value of the penalty parameter $\rho > 0$, the following ratios corresponding to the j-th subproblem solver iteration:

$$r_v^{(j)} := \frac{J^{(0)} - J(d^{(j)};0)}{J^{(0)} - D(\mathbf{w}^{(j)};0)} \quad \text{and} \quad r_\phi^{(j)} := \frac{J^{(0)} - J(d^{(j)};\rho^{(j)})}{J^{(0)} - D(\mathbf{u}^{(j)};\rho^{(j)})}. \tag{5.3.3}$$

The critical property of these ratios is that, if they are sufficiently large, then the corresponding subproblem solver iterate must yield a reduction in the penalty function model that is proportional to that yielded by an exact subproblem solution. In particular, suppose that for some prescribed constant $\beta_v \in (0,1)$ we have

$$r_v^{(j)} \geq \beta_v. \tag{R_v}$$

We may then observe that the reduction in the linearized constraint violation model $J(\cdot;0)$ (relative to a zero step) yielded by the subproblem solver iterate $d^{(j)}$ satisfies

$$\begin{aligned} J^{(0)} - J(d^{(j)};0) &\geq \beta_v(J^{(0)} - D(\mathbf{w}^{(j)};0)) \\ &\geq \beta_v(J^{(0)} - D(\mathbf{u}_0^*;0)) \geq \beta_v(J^{(0)} - J(d_0^*;0)), \end{aligned}$$

where the first inequality follows by (R_v), the second follows by optimality of $\mathbf{u}_0^*$ with respect to the dual of the feasibility subproblem, and the last follows by weak duality.

Similarly, if for some prescribed constant $\beta_\phi \in (0,1)$ and $\rho > 0$ we have

$$r_\phi^{(j)} \geq \beta_\phi, \tag{R_ϕ}$$

then it follows that

$$\begin{aligned} J^{(0)} - J(d^{(j)};\rho^{(j)}) &\geq \beta_\phi(J^{(0)} - D(\mathbf{u}^{(j)};\rho^{(j)})) \\ &\geq \beta_\phi(J^{(0)} - D(\mathbf{u}^*_{\rho^{(j)}};\rho^{(j)})) \geq \beta_\phi(J^{(0)} - J(d^*_{\rho^{(j)}};\rho^{(j)})). \end{aligned} \tag{5.3.4}$$

Our penalty parameter strategy is motivated by the desire to ensure that if the j-th iterate of the subproblem solver represents a sufficiently accurate solution of the penalty subproblem for $\rho > 0$, then it should also represent a sufficiently accurate solution of the feasibility subproblem; otherwise, then the penalty parameter should be reduced. Specifically, choosing parameters

$$0 < \beta_v < \beta_\phi < 1, \tag{5.3.5}$$

we initialize $\rho^{(0)} \leftarrow \rho^{k-1}$ (from the preceding iteration of the penalty-SQO framework) and apply the subproblem solver to (QP_ρ) to initialize the sequence $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$. If, at the start of the j-th subproblem solver iteration we have that (R_ϕ) is not satisfied, then we continue the iteration to solve (QP_ρ) for the current value of ρ. Otherwise, if (R_ϕ) is satisfied but (R_v) is not, then we reduce the penalty parameter by setting

$$\rho^{(j)} \leftarrow \theta_\rho \rho^{(j-1)} \tag{5.3.6}$$

for some prescribed constant $\theta_\rho \in (0,1)$. (The remaining case is that (R_ϕ) and R_v are both satisfied, in which case we do not change the penalty parameter and may either terminate the subproblem solver or continue to compute a more accurate solution of (QP_ρ). The determination of whether to terminate or continue the subproblem solver should be made based on the demands of the penalty-SQO method.)

We summarize our $d$ynamic $u$pdating strategy below:

$$\boxed{\text{For } (d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)}), \text{ if } (R_\phi) \text{ holds but } (R_v) \text{ does not, then apply (5.3.6).}} \quad \text{(DUST)}$$

We close this subsection by making a few practical remarks regarding the use of (DUST) within a subproblem solver for (QP$_\rho$). In particular, while we have defined the sequence $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$ as being generated by (a single run of) the solver, it may be reasonable to reinitialize the solver—or at least perform some auxiliary computations—after any iteration in which (5.3.6) is invoked. (Such auxiliary computations may involve scaling vectors and/or matrices due to the change in the penalty parameter; e.g., see the discussion of the Hessian approximation strategy in §5.5.3.) That being said, it is reasonable to assume that, during any sequence of iterations in which ρ does not change, the subproblem solver would be applied as if it were being applied to a (static) instance of (QP$_\rho$). In such a manner, any convergence guarantees for the subproblem solver would hold if/when the penalty parameter stabilizes at a fixed value, as is guaranteed to occur in certain situations described next.

5.3.3 Finite Updates

The purpose of this subsection is to show that if (DUST) is employed within an algorithm for solving (QP$_\rho$), then, under reasonable assumptions on the subproblem data, for any $\rho^{(j)} \in (0, \tilde{\rho}]$ for some sufficiently small $\tilde{\rho} > 0$ whose value depends only on the subproblem data, if (R_ϕ) is satisfied, then (R_v) is also satisfied. In other words, after finite iterations, the updating strategy (5.3.6) will never be triggered. Let $\underline{\lambda}_0$ and $\bar{\lambda}_0$ be the smallest and largest eigenvalues of H_0, and so forth for $\underline{\lambda}_\rho$ and $\bar{\lambda}_\rho$ for H_ρ. Notice that

$$\underline{\lambda}_{\rho^{(j)}} \geq \min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0) \quad \text{and} \quad \bar{\lambda}_{\rho^{(j)}} \leq \max(\bar{\lambda}_{\rho^{(0)}}, \bar{\lambda}_0). \tag{5.3.7}$$

We formalize our assumptions for this analysis as the following.

Assumption 5.3.2. *Let $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$ be a sequence such that, for all $j \in \mathbb{N}$, the* *vectors* $\mathbf{u}^{(j)}$ *and* $\mathbf{w}^{(j)}$ *are feasible for* (DQP$_\rho$). *Moreover, let* $\{\rho^{(j)}\}$ *be a sequence generated along with* $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$ *by applying* (DUST). *Then, corresponding to the set*

$$\mathcal{U} = \{j : (d^{(j)}, \mathbf{u}^{(j)}) \text{ satisfies } (R_\phi)\},$$

the subsequences $\{\|\mathbf{u}^{(j)}\|_2\}_{k\in\mathcal{U}}$ *and* $\{\|\mathbf{w}^{(j)}\|_2\}_{k\in\mathcal{U}}$ *are bounded by a constant* $\kappa_0 > 0$ *independent of* $\{\rho^{(j)}\}$.

The boundedness assumption on dual estimates are reasonable since our subproblems are assumed to be strictly convex. We can also easily show the primal variables $\{d^{(j)}\}_{j\in\mathcal{U}}$ are also bounded.

Lemma 5.3.1. *If Assumption 5.3.1 and 5.3.2 hold, then we know*

$$\|d^{(j)}\|_2 \le \kappa_1 := \frac{\rho^{(0)}\|g\|_2 + \sqrt{(\rho^{(0)})^2\|g\|_2^2 + 4\max(\bar{\lambda}_{\rho^{(0)}}, \bar{\lambda}_0)J^{(0)}}}{2\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)}, \quad j \in \mathcal{U}. \tag{5.3.8}$$

Proof. Notice by Assumption 5.3.2 (i), it holds true that $\{d^{(j)}\}_{j\in\mathcal{U}} \subset X$, which implies $\delta(d^{(j)}|X) = 0$ for $j \in \mathcal{U}$. By (R_ϕ), every $(d^{(j)}, \mathbf{u}^{(j)}, \rho^{(j)})$ must satisfies (5.3.4), which indicates

$$J(d^{(j)}; \rho^{(j)}) = \rho^{(j)} g^T d^{(j)} + \tfrac{1}{2}(d^{(j)})^T H_{\rho^{(j)}} d^{(j)} \le J^{(0)}.$$

It follows that

$$\underline{\lambda}_{\rho^{(j)}}\|d^{(j)}\|_2^2 \le J^{(0)} + |\rho^{(j)} g^T d^{(j)}| \le J^{(0)} + \rho^{(0)}\|g\|_2\|d^{(j)}\|_2.$$

This implies

$$\|d^{(j)}\|_2 \le \frac{\rho^{(0)}\|g\|_2 + \sqrt{(\rho^{(0)})^2\|g\|_2^2 + 4\underline{\lambda}_{\rho^{(j)}}J^{(0)}}}{2\underline{\lambda}_{\rho^{(j)}}J^{(0)}},$$

which, together with (5.3.7), proves (5.3.8). □

The next lemma shows the difference between the primal values of optimality subproblem and feasibility subproblem at iteration $j \in \mathcal{U}$ is dependent of ρ, and so is the difference between dual values.

Lemma 5.3.2. *There exists a constant $\kappa_3 >$ independent of j such that for any $j \in \mathcal{U}$,*

$$|J(d^{(j)};\rho^{(j)}) - J(d^{(j)};0)| \leq \kappa_2 \rho^{(j)}, \tag{5.3.9a}$$

$$|D(\mathbf{u}^{(j)};\rho^{(j)}) - D(\mathbf{w}^{(j)};0)| \leq \kappa_3 \rho^{(j)}, \tag{5.3.9b}$$

with

$$\begin{aligned}
\kappa_2 &= \tfrac{1}{2}\|H_f\|_2 \kappa_1^2 + \|g\|_2 \kappa_1, \\
\kappa_3 &= \frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)}(\|H_0^{-1}H_f\|_2 + \tfrac{1}{2}\|g\|_2) + \|H_0^{-1}H_f\|_2\|g\|_2.
\end{aligned}$$

Proof. For primal value, it holds true that

$$\begin{aligned}
|J(d^{(j)};\rho^{(j)}) - J(d^{(j)};0)| &= |\rho^{(j)} g^T d^{(j)} + \tfrac{1}{2}(d^{(j)})^T H_{\rho^{(j)}} d^{(j)} - \tfrac{1}{2}(d^{(j)})^T H_0 d^{(j)}| \\
&= |\rho^{(j)} g^T d^{(j)} + \tfrac{\rho^{(j)}}{2}(d^{(j)})^T H_f d^{(j)}| \\
&\leq \rho^{(j)}(\|g\|_2 + \tfrac{1}{2}\|H_f\|_2\|d^{(j)}\|_2)\|d^{(j)}\|_2,
\end{aligned} \tag{5.3.10}$$

which combined with Lemma 5.3.1 immediately proves (5.3.9a).

Let $\hat{y}^{(j)} = H_{\rho^{(j)}}^{-1}(u_0^{(j)} - \rho^{(j)} g)$ and $\bar{y}^{(j)} = H_0^{-1} u_0^{(j)}$. By Lemma 5.3.1, one can show

$$\|\hat{y}^{(j)}\|_2 \leq \frac{\kappa_0 + \rho^{(j)}\|g\|_2}{\underline{\lambda}_{\rho^{(j)}}} \leq \frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)},$$

and $\{\|\bar{y}^{(j)}\|_2\}_{j\in\mathcal{U}}$ are all bounded by constant

$$\|\bar{y}^{(j)}\|_2 \leq \frac{\kappa_0}{\underline{\lambda}_0}.$$

It follows that

$$\rho^{(j)} g = u_0^{(j)} - (u_0^{(j)} - \rho^{(j)} g) = H_0 \bar{y}^{(j)} - H_{\rho^{(j)}} \hat{y}^{(j)} = H_0(\bar{y}^{(j)} - \hat{y}^{(j)}) - \rho^{(j)} H_f \hat{y}^{(j)},$$

which implies for any $j \in \mathcal{U}$,

$$\begin{aligned}\|\bar{y}^{(j)} - \hat{y}^{(j)}\|_2 &= \|\rho^{(j)} H_0^{-1} H_f (\hat{y}^{(j)} + g)\|_2 \\ &\le \rho^{(j)} \|H_0^{-1} H_f\|_2 \|\hat{y}^{(j)} + g\|_2 \\ &\le \rho^{(j)} \|H_0^{-1} H_f\|_2 \left(\frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)} + \|g\|_2 \right).\end{aligned} \tag{5.3.11}$$

The difference between the dual values of optimality subproblem and feasibility subproblem is then given by

$$\begin{aligned}&|D(\mathbf{u}^{(j)};\rho^{(j)}) - D(\mathbf{u}^{(j)};0)| \\ =\ & | - \tfrac{1}{2}(u_0^{(j)} - \rho^{(j)} g)^T H_{\rho^{(j)}}^{-1} (u_0^{(j)} - \rho^{(j)} g) + \tfrac{1}{2}(u_0^{(j)})^T H_0^{-1} u_0^{(j)}| \\ =\ & |\tfrac{1}{2}(\bar{y}^{(j)} - \hat{y}^{(j)})^T u_0^{(j)} + \tfrac{1}{2}\rho^{(j)} g^T \hat{y}^{(j)}| \\ \le\ & \tfrac{1}{2}\|\bar{y}^{(j)} - \hat{y}^{(j)}\|_2 \|u_0^{(j)}\|_2 + \tfrac{1}{2}\rho^{(j)}\|g\|_2\|\hat{y}^{(j)}\|_2 \\ \le\ & \rho^{(j)} \left(\|H_0^{-1} H_f\|_2 \left(\frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)} + \mathbb{N}orm{g}_2 \right) + \tfrac{1}{2}\|g\|_2 \frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)} \right) \\ =\ & \rho^{(j)} \left(\frac{\kappa_0 + \rho^{(0)}\|g\|_2}{\min(\underline{\lambda}_{\rho^{(0)}}, \underline{\lambda}_0)} (\|H_0^{-1} H_f\|_2 + \tfrac{1}{2}\|g\|_2) + \|H_0^{-1} H_f\|_2 \|g\|_2 \right),\end{aligned}$$

where the last inequality is by (5.3.11) and Lemma 5.3.1. This completes the proof of (5.3.9b). □

Now we are ready to prove our main result.

Theorem 5.3.3. *Given $g, H_f, H_0, \rho^{(j)}$ and set C satisfying Assumption 5.3.1. Consider an algorithm for solving* (QP$_\rho$) *such that Assumption 5.3.2 is satisfied. Let*

$$\tilde{\rho} := \frac{1 - \sqrt{\beta_v/\beta_\phi}}{\kappa_3} (J^0 - D(\mathbf{u}^{(j)};0)).$$

For (QP$_\rho$) *with any $\rho^{(j)} \in (0, \tilde{\rho})$, if the primal-dual iterate $\{d^{(j)}, \mathbf{u}^{(j)}\}$ generated by the algorithm satisfies condition R_ϕ, then $\{d^{(j)}, \mathbf{w}^{(j)}\}$ also satisfy R_v. In other words, for any $\rho^{(j)} \in (0, \tilde{\rho})$, the* (DUST) *is never triggered.*

Proof. By Lemma 5.3.2, we can see that for problem (QP_ρ) with sufficiently small $\rho^{(j)} > 0$, the primal-dual iterates in $\mathcal{U}$ generated by the given algorithm satisfy

$$\frac{J^0 - J(d^{(j)};0)}{J^0 - J(d^{(j)};\rho^{(j)})} > \sqrt{\frac{\beta_v}{\beta_\phi}}, \tag{5.3.12a}$$

$$\frac{J^0 - D(\mathbf{u}^k;\rho^{(j)})}{J^0 - D(\mathbf{w}^k;0)} \geq \frac{J^0 - D(\mathbf{u}^k;\rho^{(j)})}{J^0 - D(\mathbf{u}^k;0)} > \sqrt{\frac{\beta_v}{\beta_\phi}}, \tag{5.3.12b}$$

by the fact that $\frac{\beta_v}{\beta_\phi} \in (0,1)$. This implies that for $j \in \mathcal{U}$

$$\frac{r_v^{(j)}}{r_\phi^{(j)}} = \frac{J^0 - J(d^{(j)};0)}{J^0 - J(d^{(j)};\rho^{(j)})} \frac{J^0 - D(\mathbf{u};\rho^{(j)})}{J^0 - D(\mathbf{u}^{(j)};0)} \geq \frac{\beta_v}{\beta_\phi},$$

yielding

$$r_v^{(j)} \geq \frac{\beta_v}{\beta_\phi} r_\phi^{(j)} \geq \beta_v, \qquad j \in \mathcal{U}$$

where the last inequality is from the fact that $r_\phi^{(j)} \geq \beta_\phi, j \in \mathcal{U}$ by the definition of set $\mathcal{U}$.

Now we determine the values of ρ that guarantee (5.3.12b). For any $j \in \mathcal{U}$, we have from (5.3.9b)

$$-\kappa_3\rho^{(j)} \leq D(\mathbf{u}^{(j)};\rho^{(j)}) - D(\mathbf{u}^{(j)};0) \leq \kappa_3\rho^{(j)},$$

which implies that

$$1 - \frac{\kappa_3\rho^{(j)}}{J^0 - D(\mathbf{u}^{(j)};0)} \leq \frac{J^0 - D(\mathbf{u}^{(j)};\rho^{(j)})}{J^0 - D(\mathbf{u}^{(j)};0)} \leq 1 + \frac{\kappa_3\rho^{(j)}}{J^0 - D(\mathbf{u}^{(j)};0)}.$$

It follows that (5.3.12b) holds true for any

$$\rho^{(j)} \leq \frac{1 - \sqrt{\beta_v/\beta_\phi}}{\kappa_3}(J^0 - D(\mathbf{u}^{(j)};0)).$$

Therefore, for any $\rho \in [0, \tilde{\rho}]$, the (DUST) is never triggered. □

5.4 SQO Subproblem Solvers

In this section, we will present matrix-free solvers with properties satisfying Assumption 5.3.2 so that DUST can be incorporated in the proposed algorithms. The first algorithm is an Alternating Direction Augmented Lagrangian (ADAL) method, also known as alternating direction method of multipliers (ADMM). It is designed for solving (QP$_\rho$) with generic convex set X. The second solver is Coordinate Descent Algorithm (CDA), which solves (QP$_\rho$) with $X = \mathbb{R}^n$.

5.4.1 An Alternating Direction Augmented Lagrangian Method

The ADAL algorithm decouples the quadratic function $\psi(\cdot\,;\rho)$ and the nonsmooth function $\text{dist}\,(\cdot \mid C\,)$ by reformulating problem (QP$_\rho$) into an equality constrained problem

$$\min_{x,p}\ \tilde{J}(x,p;\rho) \quad \text{such that } x = p_i, \quad i = 1,\ldots,m,$$

where

$$\tilde{J}(x,p;\rho) := \varphi(x;\rho) + \text{dist}\,(p \mid C\,) + \delta(x|X),$$

and auxiliary variables $p = [p_1^T,\ldots,p_m^T]^T$ with $p_i \in \mathbb{R}^{d_i}$. Define dual variable v_i corresponding to the constraint $x = p_i$, $i = 1,\ldots,m$, and $v = [v_1^T,\ldots,v_m^T]^T$. The augmented Lagrangian for (QP$_\rho$), with penalty parameter $\mu > 0$ (we call μ as the subproblem penalty parameter), is given by

$$L_\mu(x,p,v;\rho) := \tilde{J}(x,p;\rho) + \frac{1}{2\mu}\sum_{i=1}^{m}\Big(\,\|x - p_i + \mu v_i\|_2^2 - \frac{\mu}{2}\,\|v_i\|_2^2\Big).$$

The iteration of ADAL method then can be stated by Algorithm 3.

Algorithm 3 ADAL method for penalty-SQO subproblems

1: (Initialization) Given μ, x^0 and v^0, set $k = 0$.

2: (Update p) Set

$$p^{k+1} \leftarrow \min_p L_\mu(x^k, p, v^k; \rho). \tag{5.4.1}$$

3: (Update x) Set

$$x^{k+1} \leftarrow \min_x L_\mu(x, p^{k+1}, v^k; \rho). \tag{5.4.2}$$

4: (Update v) Set

$$v_i^{k+1} \leftarrow v_i^k + \frac{1}{\mu}(x^{k+1} - p_i^{k+1}), \quad \text{for} \quad i = 1, \ldots, m. \tag{5.4.3}$$

5: Set $k \leftarrow k + 1$.

Define $y_i^k = x^k + \mu u_i^k$. The solution of (5.4.1) can be calculated explicitly by

$$p_i^{k+1} = \begin{cases} P_{C_i}(y_i^k) & \text{if } \operatorname{dist}_2\left(y_i^k \mid C_i\right) \leq \mu, \\ y_i^k - \frac{\mu}{\operatorname{dist}_2\left(y_i^k \mid C_i\right)}(y_i^k - P_{C_i}(y_i^k)) & \text{if } \operatorname{dist}_2\left(y_i^k \mid C_i\right) > \mu. \end{cases}$$

Subproblem (5.4.2) is a constrained QO problem. We can write $L_\mu(x, p^{k+1}, u^k; \rho)$ equivalently as

$$L_\mu(x, p^{k+1}, u^k; \rho) = \frac{1}{2}x^T H_{\rho,\mu} x - (\tilde{g}^k)^T x + \delta(x|X)$$

by omitting the constant term, where

$$H_{\rho,\mu} = H_\rho + \frac{m}{\mu} I$$
$$\tilde{g}^k = -\rho g + \frac{1}{\mu}\sum_{i=1}^{m}(p_i^{k+1} - \mu v_i^k).$$

Notice that we use different notation v for the dual variables. This is because the dual iterates generated by ADAL may not be dual feasible. Now we discuss approaches to construct dual feasible multipliers at each iteration. The first approach derives $\{u_i^k\}_{i=1}^m$

by projecting $\{v^k\}$ onto the dual feasible region:

$$u_i^k = \eta_i^k \bar{a}_i \tag{5.4.4}$$

where

$$\eta_i^k = \begin{cases} \text{mid}\{-1, (v_i^k)^T \bar{a}_i, 1\}, & i = 1, \ldots, s, \\ \text{mid}\{\ \ 0, (v_i^k)^T \bar{a}_i, 1\}, & i = s+1, \ldots, m. \end{cases}$$

An alternative approach is to set

$$u_i^k = v_i^k - \frac{1}{\mu}(x^k - x^{k-1}), \quad i = 1, ..., m. \tag{5.4.5}$$

We know from [11, Lemma 3.2] that estimate (5.4.5) is also dual feasible. At iteration $k-1$, we know by the KKT condition of (5.4.2) that $-H_{\rho,\mu}x^k + \tilde{g}^k \in N(x^k|X)$. We can set

$$u_{m+1}^k = -H_{\rho,\mu}x^k + \tilde{g}^k, \tag{5.4.6}$$

so that $\delta^*(u_{m+1}^k|X) = \langle x^k, u_{m+1}^k \rangle$. Accordingly, we have

$$u_0^k = -\sum_{i=1}^{m+1} u_i^k. \tag{5.4.7}$$

To incorporate DUST, Assumption 5.3.2 must be satisfied by the proposed ADAL algorithm. We only have to show the boundedness of $\{u_{m+1}^k\}_{j\in\mathcal{U}}$, which is summarized in the following proposition.

Proposition 5.4.1. *Assumption 5.3.2 is satisfied by Algorithm 3 with $\{u_{m+1}^k\}$ set by* (5.4.6)*, i.e., $\{\|u_{m+1}^k\|_2\}_{j\in\mathcal{U}}$ are bounded by some constant $\kappa > 0$ independent of k and ρ.*

Proof. Let $\Omega := \{v^\rho | \rho \in [0, \rho^0]\}$ where (x^ρ, p^ρ, v^ρ) is the optimal primal-dual solution of $\tilde{J}(x, p; \rho)$. By optimality condition $v_i^\rho \in \partial\text{dist}_2\,(p_i^\rho \mid C_i)$, Ω is bounded by some positive constant independent of ρ. From [11, Lemma 3.5], we know $\{\|v^k - v^\rho\|_2\}$ is uniformly bounded for a fixed ρ. Therefore, overall we have $\{\|v^k\|_2\}$ are bounded by some positive constant independent of k and ρ. This result, combined with [11, Lemma 3.2], implies $\{\tilde{g}^k\}$ is bounded by some constant $\kappa_4 > 0$ independent of k and ρ. Notice that $\{\|H_\rho\|_2 | \rho \in [0, \rho^0]\}$ are bounded by some constant κ_5 independent of k and ρ. By Lemma 5.3.1, we know $\{\|u_{m+1}^k\|_2\}_{j\in\mathcal{U}}$ are bounded by $\kappa := \kappa_5\kappa_1 + \kappa_4$. $\square$

5.4.2 A Coordinate Descent Method

In this section, we consider combining the penalty parameter updating with coordinate descent method. We have the following two problems of interest.

$$\min_{x\in\mathbb{R}^n} J(x;\rho) := \frac{1}{2}x^T H_\rho x + \rho g^T x + \sum_{i=1}^{s} |a_i^T x + b_i| + \sum_{i=s+1}^{m} (a_i^T x + b_i)_+ \tag{5.4.8}$$

$$\min_{z\in\mathbb{R}^n} J(z;0) := \frac{1}{2}z^T H_0 z + \sum_{i=1}^{s} |a_i^T z + b_i| + \sum_{i=s+1}^{m} (a_i^T z + b_i)_+. \tag{5.4.9}$$

Direct computation shows the Lagrangian dual problems of (5.4.8) and (5.4.9) are respectively

$$\max_{l\le\boldsymbol{\eta}\le c} D(\boldsymbol{\eta};\rho) := -\frac{1}{2}(A^T\boldsymbol{\eta} - g)^T H_\rho^{-1}(A^T\boldsymbol{\eta} - g) + \boldsymbol{\eta}^T\boldsymbol{b} \tag{5.4.10}$$

$$\max_{l\le\boldsymbol{\lambda}\le c} D(\boldsymbol{\lambda};0) := -\frac{1}{2}\boldsymbol{\lambda}^T A H_0^{-1} A^T\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{b} \tag{5.4.11}$$

where $l = [-\mathbf{1}_s^T, \mathbf{0}_{m-s}^T]^T$ and $c = \mathbf{1}_m$. The solutions to (5.4.8) and (5.4.9) are recovered by $x = -H_\rho^{-1}(\rho g + A^T\boldsymbol{\eta})$ and $z = -H_0^{-1}A^T\boldsymbol{\lambda}$ respectively. At iteration k, define

$$r_v^k := \frac{J^{(0)} - J(x^k;0)}{J^{(0)} - \max\{D(\boldsymbol{\lambda}^k;0), D(\boldsymbol{\eta}^k;0)\}}$$
$$r_\phi^k := \frac{J^{(0)} - J(x^k;\rho^{k-1})}{J^{(0)} - D(\boldsymbol{\eta}^k;\rho^{k-1})}.$$

Now we are ready to present our algorithm which combines the coordinate descent with dynamic penalty parameter updating.

Algorithm 4 Coordinate Descent Method for Penalty-SQO Subproblems

(Initialization) Set $\boldsymbol{\eta}^0$, β_v, β_ϕ, $\theta_\rho \in (0,1)$ and $k = 0$.
(Update $\boldsymbol{\eta}^k$ and $\boldsymbol{\lambda}^k$)
for $i = 1, \ldots, m$ **do**

$$\boldsymbol{\eta}_i^k := \underset{l_i \le \eta_i \le c_i}{\operatorname{argmin}} D(\eta_1^k, \ldots, \eta_{i-1}^k, \eta_i, \eta_{i+1}^{k-1}, \ldots, \eta_m^{k-1}; \rho^{k-1}) \tag{5.4.12}$$

$$\boldsymbol{\lambda}_i^k := \underset{l_i \le \lambda_i \le c_i}{\operatorname{argmin}} D(\lambda_1^k, \ldots, \lambda_{i-1}^k, \lambda_i, \lambda_{i+1}^{k-1}, \ldots, \lambda_m^{k-1}; 0) \tag{5.4.13}$$

    Update $x^k := -H_\rho^{-1}(\rho g + A^T \boldsymbol{\eta}^k)$, r_ϕ^k and r_v^k.
    (Update ρ^k)
    If $r_\phi^k > \beta_\phi$ and $r_v^k < \beta_v$, set $\rho^k \leftarrow \theta_\rho \rho^{k-1}$.
    Set $k \leftarrow k + 1$.
end for

5.5 Implementation

In this section, we discuss the implementation issues of subproblem solvers when applied to different scenarios.

5.5.1 Inexact Solution

In this subsection, we discuss a strategy to terminate when an inexact solution is found. Generally, the subproblem (QP$_\rho$) need not to be solved accurately. An inexact solution may be accepted to guarantee the global convergence as long as it achieves sufficient improvement on the penalty function. We describe a termination criterion based on the ratio r_ϕ^k defined in subsection 5.3.2. At each iteration, if condition R_ϕ and R_v are both satisfied, we terminate the algorithm. By doing this, the algorithm ends up with an inexact solution for (QP$_\rho$), along which the duality gap for optimality problem and feasibility problem can be both reduced by a fraction.

5.5.2 Adding A Trust Region

If problem (QP_ρ) is unconstrained ($X = \mathbb{R}^n$), the solution of ADAL subproblem (5.4.2) can be given by

$$x^{k+1} = H_{\rho,\mu}^{-1}\, \tilde{g}^k.$$

The proposed ADAL algorithm can also handle the case where X is a trust region with

$$X = \{x \mid \|x\|_2 \leq \Delta\}\,.$$

In this case, the solution to (5.4.2) is given by

$$x_\eta := (\eta I + H_{\rho,\mu})^{-1}\tilde{g}^k,$$

where η is any solution to the equation

$$\phi(\eta) = 0,$$

with

$$\phi(\eta) := \frac{1}{\Delta} - \frac{1}{\|x_\eta\|_2} \tag{5.5.1}$$

on the interval $(\bar{\eta}, +\infty)$. Here $\bar{\eta}$ is the smallest eigenvalue of $H_{\rho,\mu}$. Since

$$\phi'(\eta) = \frac{\tilde{g}^T(\eta I + H_{(\rho,\mu)})^{-3}\tilde{g}}{\|x_\eta\|_2^3},$$

applying Newton's method to this equation requires the solving systems of the form

$$(\eta I + H_{(\rho,\mu)})x = -\tilde{g} \quad \text{and} \quad (\eta I + H_{(\rho,\mu)})^{-2}x = -\tilde{g}$$

for a few values of η.

5.5.3 Low-Rank Approximation

We describe an implementation when using low-rank approximation for the Hessian H_f and H_0. Assume the Hessian takes the decomposition

$$\begin{aligned} H_\rho &= \sigma I + \Psi \Sigma^{-1} \Psi^T, \\ H_0 &= \gamma I + \Phi \Gamma^{-1} \Phi^T, \end{aligned}$$

where $\Psi \in \mathbb{R}^{n \times r}$ with $r \ll n$, and $\Phi \in \mathbb{R}^{n \times l}$ with $l \ll n$ are low rank matrices, and $\Sigma \in \mathbb{R}^{r \times r}, \Gamma \in \mathbb{R}^{l \times l}$ are invertible. We investigate the inverse of H_ρ and $H_{\rho,\mu}$ by using the following generalized matrix inversion formula. For any given invertible $A \in \mathbb{R}^{n \times n}$, invertible $S \in \mathbb{R}^{l \times l}$, and $U, V \in \mathbb{R}^{n \times l}$, the following Sherman Morrison formula holds

$$(A + USV^T)^{-k} = A^{-k} - A^{-k} U (S^{-1} + V^T A^{-k} U)^{-1} W V^T A^{-k}, \tag{5.5.2}$$

where

$$W = \sum_{j=0}^{k-1} (S^{-1} (S^{-1} + V^T A^{-k} U)^{-1})^{(j)}.$$

In particular,

$$(A + USV^T)^{-1} = A^{-1} - A^{-1} U (S^{-1} + V^T A^{-1} U)^{-1} V^T A^{-1}. \tag{5.5.3}$$

Let

$$H_{\rho,\mu} = H_\rho + \frac{m}{\mu} I = (\sigma + \frac{m}{\mu}) I + \Psi \Sigma^{-1} \Psi^T.$$

Using (5.5.3), the inverse of H_0, H_ρ and $H_{\rho,\mu}$ can be computed by

$$\begin{aligned} H_0^{-1} &= \frac{1}{\gamma} \left[I - \Phi (\gamma \Gamma + \Phi^T \Phi)^{-1} \Phi^T \right] \\ H_\rho^{-1} &= \frac{1}{\sigma} \left[I - \Psi (\sigma \Sigma + \Psi^T \Psi)^{-1} \Psi^T \right] \\ H_{\rho,\mu}^{-1} &= \frac{1}{\sigma + \frac{m}{\mu}} \left[I - \Psi [(\sigma + \frac{m}{\mu}) \Sigma + \Psi^T \Psi]^{-1} \Psi^T \right]. \end{aligned}$$

We can write them in a compact form. Define

$$\Theta^T = (\gamma\Gamma + \Phi^T\Phi)^{-1}\Phi^T$$
$$\Theta_1^T = (\sigma\Sigma + \Psi^T\Psi)^{-1}\Psi^T$$
$$\Theta_2^T = [(\sigma + \frac{m}{\mu})\Sigma + \Psi^T\Psi]^{-1}\Psi^T$$

Then

$$H_0^{-1} = \frac{1}{\gamma}(I - \Phi\Theta^T) \tag{5.5.4}$$
$$H_\rho^{-1} = \frac{1}{\sigma}\left[I - \Psi\Theta_1^T\right] \tag{5.5.5}$$
$$H_{\rho,\mu}^{-1} = \frac{1}{\sigma + \frac{m}{\mu}}\left[I - \Psi\Theta_2^T\right]. \tag{5.5.6}$$

After reducing ρ to a smaller value $\bar{\rho}$, we have

$$\begin{aligned} H_{\bar{\rho}} &= \bar{\rho}H_f + H_0 \\ &= \frac{\bar{\rho}}{\rho}(H_0 + \rho H_f) + (1 - \frac{\bar{\rho}}{\rho})H_0 \\ &= \tau H_\rho + (1-\tau)H_0 \\ &= \bar{\sigma}I + \tau\Psi\Sigma^{-1}\Psi^T + (1-\tau)\Phi\Gamma^{-1}\Phi^T \\ &= H_\tau + (1-\tau)\Phi\Gamma^{-1}\Phi^T, \end{aligned}$$

with

$$\tau = \frac{\bar{\rho}}{\rho},\ \bar{\sigma} = \tau\sigma + (1-\tau)\gamma,\ H_{\bar{\tau}} = \bar{\sigma}I + \tau\Psi\Sigma^{-1}\Psi^T.$$

So we have

$$\Theta_3^T = (\frac{\bar{\sigma}}{\tau}\Sigma + \Psi^T\Psi)^{-1}\Psi^T \tag{5.5.7}$$

$$H_{\bar{\tau}}^{-1} = \frac{1}{\bar{\sigma}}[I - \Psi\Theta_3^T] \tag{5.5.8}$$

$$\Theta_4^T = \left[\frac{1}{1-\tau}\Gamma + \Phi^T H_{\bar{\tau}}^{-1}\Phi\right]^{-1}\Phi^T \tag{5.5.9}$$

$$H_{\bar{\rho}} = H_{\bar{\tau}}^{-1} - H_{\bar{\tau}}^{-1}\Phi\Theta_4^T H_{\bar{\tau}}^{-1}. \tag{5.5.10}$$

Substituting σ with $\sigma + \frac{m}{\mu}$, and repeating (5.5.7), we have the inverse of $H_{\bar{\rho},\mu}$:

$$\tilde{\sigma} = \tau\sigma + \frac{m}{\mu} + (1-\tau)\gamma,\ \ H_{\tilde{\tau}} = \tilde{\sigma}I + \tau\Psi\Sigma^{-1}\Psi^T \tag{5.5.11}$$

$$\Theta_5^T = (\frac{\bar{\sigma}}{\tau}\Sigma + \Psi^T\Psi)^{-1}\Psi^T \tag{5.5.12}$$

$$H_{\tilde{\tau}}^{-1} = \frac{1}{\bar{\sigma}}[I - \Psi\Theta_3^T] \tag{5.5.13}$$

$$\Theta_6^T = \left[\frac{1}{1-\tau}\Gamma + \Phi^T H_{\bar{\tau}}^{-1}\Phi\right]^{-1}\Phi^T \tag{5.5.14}$$

$$H_{\bar{\rho},\mu} = H_{\tilde{\tau}}^{-1} - H_{\tilde{\tau}}^{-1}\Phi\Theta_4^T H_{\tilde{\tau}}^{-1}. \tag{5.5.15}$$

5.5.4 Coordinate Descent Implementation

Using (5.5.4), (5.4.11) can be written as

$$\max_{l\le\boldsymbol{\lambda}\le c} D(\boldsymbol{\lambda};0) := -\frac{1}{2\gamma}\boldsymbol{\lambda}^T AA^T\boldsymbol{\lambda} + \frac{1}{2\gamma}\boldsymbol{\lambda}^T A\Phi\Theta^T A^T\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{b}, \tag{5.5.16}$$

in large scale setting, we assume it is not practical to calculate and store AA^T. Usually A will have nice sparse structure, while AA^T does not. Define $Q := A\Phi$ and $\tilde{Q} := A\Theta$, then (5.5.16) becomes

$$\max_{l\le\boldsymbol{\lambda}\le c} D(\boldsymbol{\lambda};0) := -\frac{1}{2\gamma}\boldsymbol{\lambda}^T AA^T\boldsymbol{\lambda} + \frac{1}{2\gamma}\boldsymbol{\lambda}^T Q\tilde{Q}^T\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{b}. \tag{5.5.17}$$

Direction derivative of $D(\lambda; 0)$ is given by

$$\frac{\partial D(\lambda;0)}{\partial \lambda_i} := \frac{1}{\gamma}\sum_{j=1}^{m}(-a_i^T a_j + q_i^T \tilde{q}_j^T)\lambda_j + b_i, \tag{5.5.18}$$

where q_i and $\tilde{q}_i$ are the i-th row of Q and $\tilde{Q}$ respectively. Then the solution of (5.4.13) is

$$\lambda_i^k = \begin{cases} l_i \quad \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} < 0 \\ [l_i, c_i] \quad \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} = 0 \\ c_i \quad \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} > 0 \\ \text{mid}\{\frac{\gamma b_i - \sum_{j=1}^{k-1}(a_i^T a_j - q_i^T \tilde{q}_j^T)\lambda_j^k - \sum_{j=k+1}^{n}(a_i^T a_j - q_i^T \tilde{q}_j^T)\lambda_j^{k-1}}{a_i^T a_i - q_i^T \tilde{q}_i^T}, l_i, c_i\} \text{ if } a_i^T a_i - q_i^T \tilde{q}_i^T \neq 0. \end{cases} \tag{5.5.19}$$

Now we see that the main calculation for the solution of (5.4.13) is the directional derivative (5.5.18). Direct computation of (5.5.18) will take $O(n^2 + nr)$ operations. It was shown in [62] that coordinate descent method will become competitive if there is an efficient way to compute the directional derivative. Here if we keep track of the following two vectors $v := \sum_{j=1}^{n} \lambda_j a_j$ and $p := \sum_{j=1}^{n} \lambda_j \tilde{q}_j$, then the complexity of the update of the directional derivative would become $O(n+r)$ which is much better than $O(n^2+nr)$. First notice that if we have v and p for most recent λ, then

$$\frac{\partial D(\lambda;0)}{\partial \lambda_i} = \frac{1}{\gamma}(-a_i^T v + q_i^T p) + b_i,$$

i.e. given v and p, calculating (5.5.18) takes only $O(n + r)$. Next let us see how to update v and p. Assume we update λ_i^{k-1} to λ_i^k, then

$$\begin{aligned} v &\leftarrow v + (\lambda_i^k - \lambda_i^{k-1})a_i \\ p &\leftarrow p + (\lambda_i^k - \lambda_i^{k-1})\tilde{q}_i. \end{aligned} \tag{5.5.20}$$

(5.5.20) shows the update of v and p is $O(n + r)$. In summary, the total complexity for each coordinate update is $O(n + r)$. Moreover, if A is a sparse matrix with an average of n_s non zeros per row, then the complexity becomes $O(n_s + r)$.

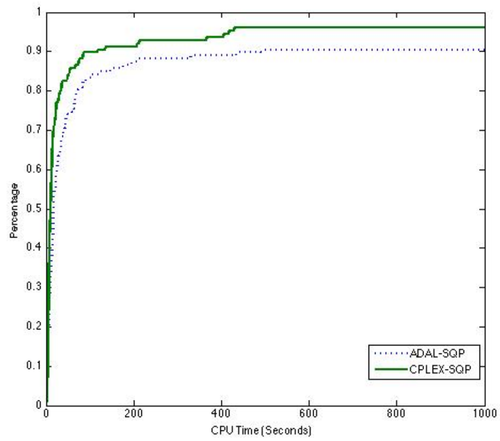
5.6 Numerical Experiments

5.6.1 ADAL

In this section, we test our proposed DUST in an ADAL subproblem solver for solving small-scale problems. We implemented Algorithm 2 and the algorithm described in §5.4.1 in MATLAB, and name it ADAL-SQO. The implementation is tested on 126 of the Hock-Schittkowski problems [49] available as AMPL models [38]. As an alternative, we also apply CPLEX as our subproblem solver. The penalty parameter is updated is updated according to DUST when an accurate solution x^k of (QP$_\rho$) is found by CPLEX. If the penalty parameter is reduced, then (QP$_\rho$) is re-solved with the new ρ and x^k as the initial point. This implementation is named as CPLEX-SQO. We use $\beta_{opt} = 0.9$ and $\beta_{fea} = 0.1$. The performance statistics is shown in Table 5.1 and the performance profile of both methods in terms of CPU time and number of iterations are shown in Figure 5.1. One can see our proposed updating strategy has acceptable performance, and overperforms the strategy of using CPLEX to accurately solve (QP$_\rho$).

Table 5.1: Performance statistics of ADAL-SQO and CPLEX-SQO

Solver	Optimal	Infeasible	Iteration
ADAL-SQO	116 (92.8%)	0	10 (7.2%)
CPLEX-SQO	116 (92.8%)	2 (1.6%)	8 (6.3%)



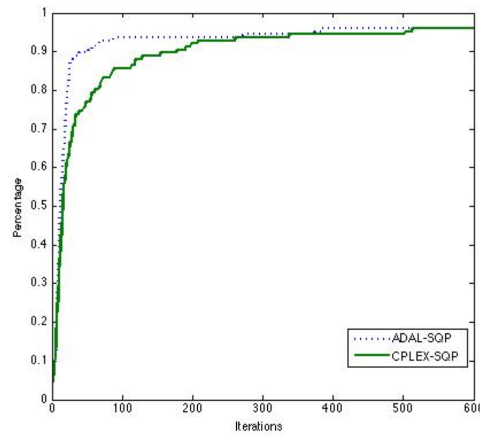


Figure 5.1: Performance profile of ADAL-SQO and CPLEX-SQO

5.6.2 Coordinate Descent

In this subsection, we test DUST on 20 Cuter problems of large scale and 107 small scale problems. Coordinate descent algorithm described in §5.4 is used to solve the subproblems. We set $\rho = 1$, $\beta_\phi = 0.7$, $\beta_v = 0.05$, $\theta_\rho = 0.5$ and the maximum iteration to be 100 for SQO algorithm. Define the maximum constraint violation $v_\infty(x)$ and the optimality error $\epsilon_\infty(x)$ as

$$v_\infty(x) := \max\{|c_i(x)| \; i = 1, \cdots, \bar{m}, (c_i(x))_+ \; i = \bar{m}+1, \cdots, m\},$$

$$\epsilon_\infty(x) := \|\nabla f(x) + \nabla c(x)\eta\|_\infty.$$

We terminate the algorithm if $v_\infty(x) \leq 10^{-7}$ and $\epsilon_\infty(x) \leq 10^{-3}$, or the consecutive improvement of constraint violation and objective function value are less than 10^{-4}.

Figure 5.2 presents the update of ρ from iteration to iteration. Table 5.2 shows the performance statistics on large scale problems and Table 5.3 shows the results on small problems. For most of the test examples, the algorithm finds the optimal solution. The final value of ρ for most test examples are not extremely small, which indicates our proposed strategy does not always quickly drive ρ to 0.

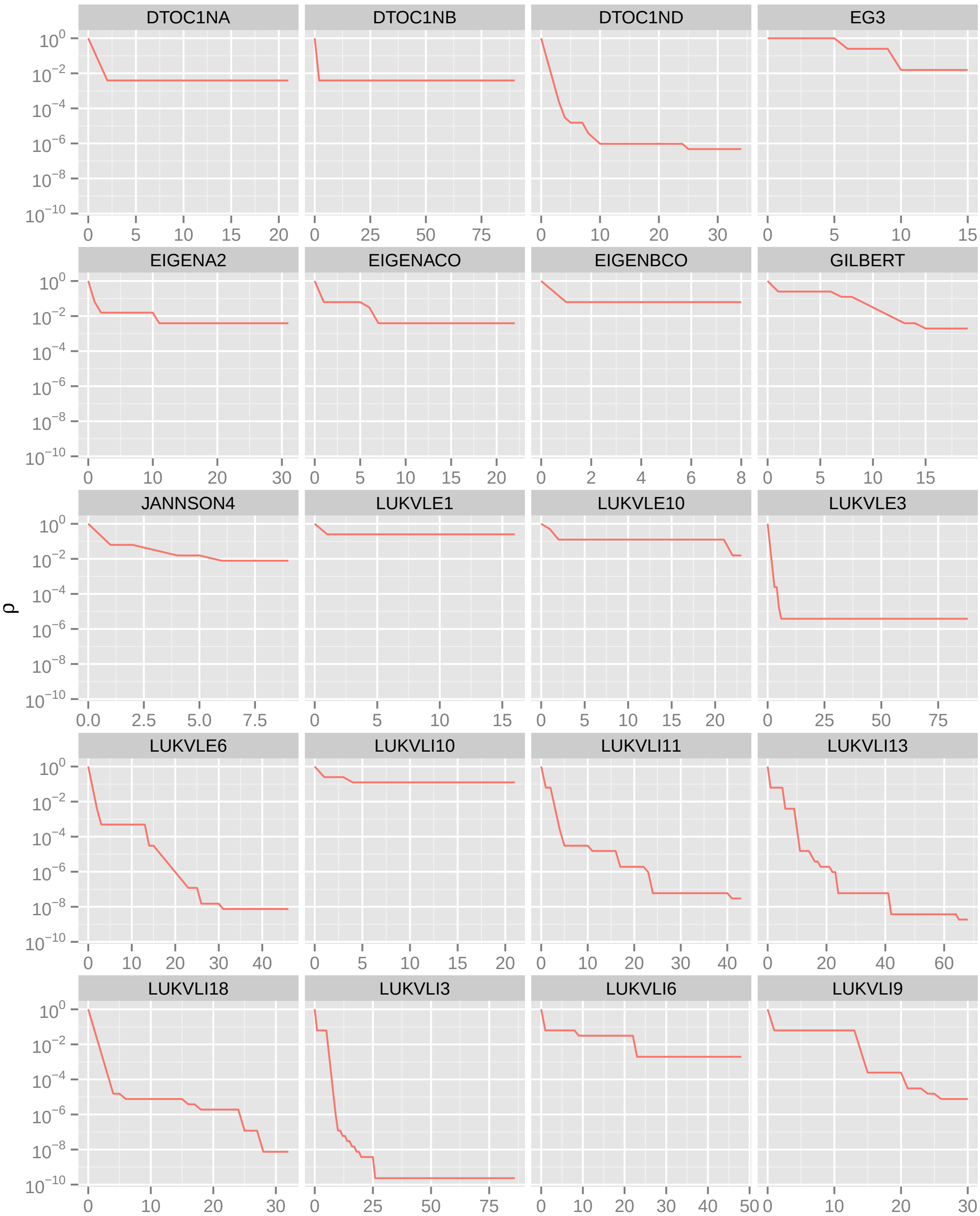


Figure 5.2: ρ update

Table 5.2: Results on 20 Cuter problem

	Name	Num_Con	Num_Eq	Num_Var	Obj	Opt_Err	ρ	Time	Violation
1	DTOC1NA	3996	3996	5998	4.1389E+00	3.8650E-05	3.9062E-03	3.6857E+00	9.9967E-08
2	DTOC1NB	3996	3996	5998	7.1388E+00	1.2129E-04	3.9062E-03	1.2981E+01	7.4492E-08
3	DTOC1ND	3996	3996	5998	4.7603E+01	5.5031E-05	4.7684E-07	5.4458E+00	1.9515E-08
4	EG3	20000	1	10001	3.4834E-04	2.6197E-02	1.5625E-02	7.0970E+01	5.2612E-10
5	EIGENA2	1275	1275	2550	3.0204E-12	3.3435E-08	3.9062E-03	1.9689E+00	0.0000E+00
6	EIGENACO	1275	1275	2550	4.5418E-09	6.2817E-07	3.9062E-03	2.1289E+00	1.7764E-15
7	EIGENBCO	1275	1275	2550	4.9000E+01	7.8237E-05	6.2500E-02	1.6198E+00	2.8207E-09
8	GILBERT	1	1	5000	2.4595E+03	3.8443E-05	1.9531E-03	5.5580E-01	9.8979E-09
9	JANNSON4	2	0	10000	9.8020E+03	2.7347E-09	7.8125E-03	1.6501E+00	4.4094E-11
10	LUKVLE1	9998	9998	10000	6.2325E+00	2.7028E-07	2.5000E-01	2.5797E+00	3.6903E-12
11	LUKVLE10	9998	9998	10000	3.5351E+03	5.6940E-04	1.5625E-02	7.4686E+00	9.0493E-08
12	LUKVLE3	2	2	10000	2.7587E+01	7.6157E-06	3.8147E-06	3.5032E+00	2.1330E-10
13	LUKVLE6	4999	4999	9999	6.2864E+05	9.2954E-06	7.4506E-09	5.1310E+00	3.0809E-12
14	LUKVLI10	9998	0	10000	3.5351E+03	2.3222E-03	1.2500E-01	4.9544E+00	2.4736E-09
15	LUKVLI11	6664	0	9998	5.0882E-05	8.3197E-04	2.9802E-08	4.7913E+00	2.6160E-08
16	LUKVLI13	6664	0	9998	1.3219E+02	4.8303E-04	1.8626E-09	9.9030E+00	4.8814E-08
17	LUKVLI18	7497	0	9997	2.8250E-04	7.0668E-04	7.4506E-09	3.3149E+01	8.4118E-09
18	LUKVLI3	2	0	10000	6.7819E+02	9.2782E-06	2.3283E-10	3.9321E+00	4.4420E-12
19	LUKVLI6	4999	0	9999	6.2864E+05	2.2232E-06	1.9531E-03	3.0957E+01	5.8365E-12
20	LUKVLI9	6	0	10000	9.9893E+02	3.9280E-04	7.6294E-06	1.6746E+00	0.0000E+00

Table 5.3: Results on 107 Cuter problems

	Name	Num_Con	Num_Eq	Num_Var	Obj	Opt_Err	Rho	Time	Violation
1	AIRPORT	42	0	84	4.7953e+04	1.3950e-02	3.0518e-05	2.3006e-01	3.6113e-10
2	ALJAZZAF	1	1	1000	3.7439e+04	2.0213e-06	7.8125e-03	6.8390e-01	2.0107e-14
3	ALLINITA	4	2	4	2.8684e+01	1.5524e-04	1.2500e-01	7.2233e-02	6.8748e-08
4	ALSOTAME	1	1	2	8.2085e-02	6.3862e-04	3.9062e-03	2.5290e-02	4.7503e-08
5	BT1	1	1	2	-1.0000e+00	4.3737e-05	1.8626e-09	4.6707e-02	4.8406e-14
6	BT11	3	3	5	8.2489e-01	1.6904e-04	1.2500e-01	4.9819e-02	3.9628e-08
7	BT12	3	3	5	6.1881e+00	1.6299e-07	2.5000e-01	3.2483e-02	6.8876e-08
8	BT2	1	1	3	3.2633e-02	1.4950e-02	3.9062e-03	5.5410e-02	7.5071e-08
9	BT4	2	2	3	-4.5511e+01	3.5266e-05	3.9062e-03	3.2753e-02	2.4067e-09
10	BT5	2	2	3	9.6172e+02	2.1414e-04	1.5625e-02	4.3901e-02	4.9495e-08
11	BT6	2	2	5	2.7704e-01	5.1709e-06	1.1921e-07	6.0202e-02	2.6437e-09
12	BT7	3	3	5	3.0650e+02	4.8483e-06	1.5625e-02	6.6016e-02	1.7396e-08
13	BT8	2	2	5	1.0000e+00	4.2395e-04	3.1250e-02	2.2081e-02	5.9359e-09
14	CHANDHEQ	100	100	100	0.0000e+00	1.7969e-04	1.0000e+00	3.3283e-01	4.6096e-08
15	DIPIGRI	4	0	7	6.8063e+02	3.3466e-04	4.8828e-04	3.1236e-01	1.9433e-09
16	DIXCHLNG	5	5	10	4.2749e+02	3.9897e-06	1.5625e-02	8.4243e-02	3.6375e-08
17	DTOC1NA	3996	3996	5998	4.1389e+00	3.8650e-05	3.9062e-03	3.6857e+00	9.9967e-08
18	DTOC1NB	3996	3996	5998	7.1388e+00	1.2129e-04	3.9062e-03	1.2981e+01	7.4492e-08
19	DTOC1ND	3996	3996	5998	4.7603e+01	5.5031e-05	4.7684e-07	5.4458e+00	1.9515e-08
20	EG3	20000	1	10001	3.4834e-04	2.6197e-02	1.5625e-02	7.0970e+01	5.2612e-10
21	EIGENA2	1275	1275	2550	3.0204e-12	3.3435e-08	3.9062e-03	1.9689e+00	0.0000e+00
22	EIGENACO	1275	1275	2550	4.5418e-09	6.2817e-07	3.9062e-03	2.1289e+00	1.7764e-15
23	EIGENBCO	1275	1275	2550	4.9000e+01	7.8237e-05	6.2500e-02	1.6198e+00	2.8207e-09
24	FLT	2	2	2	4.2055e-13	9.0334e-06	1.0000e+00	4.6640e-02	3.6520e-08
25	GILBERT	1	1	5000	2.4595e+03	3.8443e-05	1.9531e-03	5.5580e-01	9.8979e-09
26	HIMMELP2	1	0	2	-6.2054e+01	6.5287e-04	5.9605e-08	1.4293e-01	0.0000e+00
27	HIMMELP3	2	0	2	-5.9013e+01	6.6207e-01	6.2500e-02	4.2135e-02	0.0000e+00
28	HIMMELP4	3	0	2	-5.9013e+01	6.6207e-01	6.2500e-02	4.1402e-02	0.0000e+00
29	HIMMELP5	3	0	2	-5.9013e+01	2.7917e-01	1.2500e-01	8.1755e-02	0.0000e+00
30	HIMMELP6	5	0	2	-5.9012e+01	9.6979e-04	1.2500e-01	5.6715e-02	0.0000e+00
31	HS100	4	0	7	6.8063e+02	7.8432e-04	4.8828e-04	2.7450e-01	4.2046e-09
32	HS100LNP	2	2	7	6.8063e+02	4.7574e-04	9.7656e-04	1.7785e-01	8.4029e-09
33	HS100MOD	4	0	7	6.7868e+02	4.1630e-04	1.2207e-04	9.7013e-02	2.3844e-08
34	HS104	5	0	8	3.9512e+00	9.7380e-05	1.9531e-03	4.7981e-01	6.2558e-09
35	HS108	13	0	9	-8.6603e-01	3.3550e-06	5.0000e-01	5.5075e-02	5.7265e-08
36	HS11	1	0	2	-8.4985e+00	5.7314e-05	3.1250e-02	4.6349e-02	9.1747e-11
37	HS113	8	0	10	2.4306e+01	1.4980e-09	6.2500e-02	2.0193e-01	8.0283e-09
38	HS12	1	0	2	-3.0000e+01	3.5007e-06	6.2500e-02	5.7973e-02	7.6328e-09
39	HS13	1	0	2	9.9175e-01	5.0040e-05	9.5367e-07	1.4292e-01	7.0685e-08
40	HS14	2	1	2	1.3935e+00	2.3381e-10	1.2500e-01	4.2471e-02	5.2969e-08
41	HS16	2	0	2	2.3145e+01	3.6807e-07	6.2500e-02	3.4749e-02	0.0000e+00
42	HS17	2	0	2	1.0000e+00	6.3146e-10	3.9062e-03	4.4620e-02	0.0000e+00
43	HS18	2	0	2	5.0000e+00	6.0053e-05	1.2500e-01	4.0808e-02	3.4408e-08
44	HS20	3	0	2	4.0199e+01	2.7355e-06	1.2500e-01	4.7416e-02	3.4262e-08
45	HS22	2	0	2	1.0000e+00	1.2487e-07	2.5000e-01	2.7152e-02	3.3388e-11
46	HS23	5	0	2	2.0000e+00	2.1838e-06	3.9062e-03	5.0496e-02	6.8905e-08
47	HS26	1	1	3	5.0373e-12	2.3155e-07	1.5625e-02	1.8848e-01	9.4935e-08
48	HS29	1	0	3	-2.2627e+01	2.7747e-06	7.8125e-03	7.3125e-02	1.3377e-10
49	HS30	1	0	3	1.0017e+00	5.7976e-02	6.2500e-02	1.2873e-01	0.0000e+00
50	HS31	1	0	3	6.0000e+00	1.6950e-04	3.1250e-02	4.5165e-02	1.0233e-10
51	HS32	2	1	3	1.0000e+00	2.2826e-04	6.2500e-02	5.7293e-02	2.6133e-08
52	HS33	2	0	3	-4.0000e+00	3.0884e-06	6.2500e-02	1.3810e-01	0.0000e+00
53	HS40	3	3	4	-2.5000e-01	1.2096e-06	5.0000e-01	4.2027e-02	7.6550e-09
54	HS42	2	2	4	1.3858e+01	7.1245e-06	7.8125e-03	5.7693e-02	5.4777e-11

	Name	Num_Con	Num_Eq	Num_Var	Obj	Opt_Err	Rho	Time	Violation
55	HS43	3	0	4	-4.4000e+01	6.8124e-05	6.2500e-02	3.6393e-02	1.4527e-08
56	HS46	2	2	5	9.3137e-07	2.5550e-05	6.2500e-02	8.9202e-02	6.7832e-09
57	HS47	3	3	5	6.9730e-10	2.2706e-06	1.5625e-02	1.3525e-01	8.0704e-08
58	HS56	4	4	7	-3.4560e+00	1.3664e-04	3.9062e-03	1.2140e-01	3.3481e-08
59	HS57	1	0	2	3.0646e-02	9.6346e-04	6.2500e-02	2.5512e-02	0.0000e+00
60	HS59	3	0	2	-7.8028e+00	3.4936e-04	1.2500e-01	4.9485e-02	0.0000e+00
61	HS6	1	1	2	3.0201e-15	6.0951e-07	1.2500e-01	4.1996e-02	1.3662e-11
62	HS60	1	1	3	3.2568e-02	5.5686e-06	1.9073e-06	7.9263e-02	6.6209e-09
63	HS61	2	2	3	-1.4365e+02	4.3415e-06	6.2500e-02	2.9176e-02	2.2308e-08
64	HS63	2	2	3	9.6172e+02	2.2026e-04	3.1250e-02	5.1150e-02	7.2822e-08
65	HS65	1	0	3	1.0444e+00	3.5713e-01	4.6566e-10	3.1992e-01	4.5364e-08
66	HS68	2	2	4	-5.0049e-01	1.1153e-01	7.4506e-09	1.3554e-01	7.2379e-08
67	HS7	1	1	2	-1.7321e+00	1.2479e-04	6.2500e-02	4.2299e-02	6.5859e-09
68	HS70	1	0	4	7.8439e-03	9.5845e-04	6.2500e-02	3.1567e-01	0.0000e+00
69	HS77	2	2	5	2.4151e-01	1.7935e-04	1.2500e-01	3.7987e-02	9.3021e-08
70	HS78	3	3	5	-2.9197e+00	1.2998e-06	1.2500e-01	4.0654e-02	5.4896e-08
71	HS79	3	3	5	7.8858e-02	8.2159e-03	3.9062e-03	9.7829e-02	3.9336e-09
72	HS80	3	3	5	5.3950e-02	5.7338e-08	6.2500e-02	1.1381e-01	5.3509e-08
73	HS81	3	3	5	5.3950e-02	1.7344e-05	6.2500e-02	1.3362e-01	1.9773e-08
74	HS83	3	0	5	-3.1026e+04	2.6364e-06	2.3842e-07	9.8275e-02	0.0000e+00
75	HS85	21	0	5	-1.2542e+00	2.9804e-02	6.2500e-02	4.0158e-02	0.0000e+00
76	HS88	1	0	2	1.3627e+00	7.3488e-05	9.7656e-04	1.0558e-01	1.6047e-14
77	HS89	1	0	3	1.3627e+00	2.2035e-04	3.9062e-03	1.2265e-01	2.3512e-09
78	HS90	1	0	4	1.3627e+00	2.6604e-04	9.7656e-04	1.2304e-01	1.3842e-09
79	HS91	1	0	5	1.3627e+00	1.1932e-02	1.9531e-03	1.0752e-01	9.3232e-11
80	HS92	1	0	6	1.3627e+00	1.6373e-03	9.7656e-04	1.5294e-01	5.2365e-13
81	JANNSON3	3	1	20000	1.9999e+04	1.6690e-03	5.0000e-01	3.2662e+00	8.4759e-10
82	JANNSON4	2	0	10000	9.8020e+03	2.7347e-09	7.8125e-03	1.6501e+00	4.4094e-11
83	LUKVLE1	9998	9998	10000	6.2325e+00	2.7028e-07	2.5000e-01	2.5797e+00	3.6903e-12
84	LUKVLE10	9998	9998	10000	3.5351e+03	5.6940e-04	1.5625e-02	7.4686e+00	9.0493e-08
85	LUKVLE3	2	2	10000	2.7587e+01	7.6157e-06	3.8147e-06	3.5032e+00	2.1330e-10
86	LUKVLE6	4999	4999	9999	6.2864e+05	9.2954e-06	7.4506e-09	5.1310e+00	3.0809e-12
87	LUKVLI10	9998	0	10000	3.5351e+03	2.3222e-03	1.2500e-01	4.9544e+00	2.4736e-09
88	LUKVLI11	6664	0	9998	5.0882e-05	8.3197e-04	2.9802e-08	4.7913e+00	2.6160e-08
89	LUKVLI12	7497	0	9997	4.3896e-03	1.5834e-02	2.3842e-07	3.8613e+00	0.0000e+00
90	LUKVLI13	6664	0	9998	1.3219e+02	4.8303e-04	1.8626e-09	9.9030e+00	4.8814e-08
91	LUKVLI17	7497	0	9997	7.8051e+02	1.9460e-02	1.8626e-09	7.0678e+01	7.7344e-09
92	LUKVLI18	7497	0	9997	2.8250e-04	7.0668e-04	7.4506e-09	3.3149e+01	8.4118e-09
93	LUKVLI3	2	0	10000	6.7819e+02	9.2782e-06	2.3283e-10	3.9321e+00	4.4420e-12
94	LUKVLI5	9996	0	10002	5.2679e-01	4.8347e-02	9.3132e-10	1.1230e+02	1.1745e-08
95	LUKVLI6	4999	0	9999	6.2864e+05	2.2232e-06	1.9531e-03	3.0957e+01	5.8365e-12
96	LUKVLI9	6	0	10000	9.9893e+02	3.9280e-04	7.6294e-06	1.6746e+00	0.0000e+00
97	MARATOS	1	1	2	-1.0000e+00	1.1276e-08	5.0000e-01	3.8167e-02	1.9167e-09
98	MATRIX2	2	0	6	1.3331e-09	5.1474e-05	1.0000e+00	9.7082e-02	2.5839e-08
99	MISTAKE	13	0	9	-1.0000e+00	3.4154e-05	5.0000e-01	8.5467e-02	8.6954e-09
100	MWRIGHT	3	3	5	2.4979e+01	4.1174e-06	1.0000e+00	1.0378e-01	9.8647e-09
101	ORTHREGB	6	6	27	7.3123e-15	8.7122e-08	1.0000e+00	1.0898e-01	3.2428e-08
102	SYNTHES1	6	0	6	7.5930e-01	9.1389e-06	6.2500e-02	6.5607e-02	1.2977e-10
103	TFI1	101	0	3	5.3347e+00	7.8091e-05	3.9062e-03	1.9285e-01	1.9879e-11
104	TWOBARS	2	0	2	1.5087e+00	1.2967e-05	2.5000e-01	7.9626e-02	3.7149e-10
105	ZECEVIC3	2	0	2	9.7309e+01	4.2420e-05	3.1250e-02	6.8378e-02	1.4316e-10
106	ZECEVIC4	2	0	2	7.5577e+00	6.3717e-02	3.1250e-02	9.0297e-02	0.0000e+00
107	ZY2	2	0	3	2.0000e+00	3.0884e-06	6.2500e-02	8.4606e-02	0.0000e+00

Chapter 6

Conclusion

In this dissertation, we have presented a SQO method named SQuID that possesses global and fast local convergence guarantees for both feasible and infeasible problems, two matrix-free solvers called IRWA and ADAL for approximately solving the exact penalty subproblem. The techniques in methods are incorporated to construct a penalty-SQO method with dynamic penalty parameter updates for solving large-scale problems.

Novelties of SQuID are its unique two-phase approach and carefully designed updating strategy for the penalty parameter. The subproblems in each phase and the penalty parameter update are designed to strike a balance between moving toward feasibility and optimality in each iteration. Near an optimal point satisfying common assumptions, the penalty parameter remains constant and the algorithm reduces to a classical SQO method, yielding fast local convergence. Similarly, near an infeasible stationary point, the penalty parameter is reduced sufficiently quickly to yield fast infeasibility detection. The convergence properties that we have proved for our algorithm were illustrated empirically on test sets of feasible and infeasible problems.

We remark, however, that there remain various practical issues that one faces when considering an implementation of SQuID. As with any SQO method, the primary concern is the efficiency of the QO subproblem solver. This is especially the case when one wishes to use exact second order derivative information and the resulting Hessian matrices are not positive definite. We have employed a Hessian modification strategy in our numerical experiments, but as for any SQO method that employs such a strategy, these modifications are cumbersome in large-scale settings and may inhibit superlinear convergence. We leave it a subject of future research to investigate ways in which inexactness can be incorporated into the subproblem solves and negative curvature can be handled, knowing that the algorithm and analysis presented in this chapter provides a strong backbone for rapid infeasibility detection when such additional features are developed.

The primary novelty of our subproblem solvers is a newly proposed IRWA for solving such problems involving arbitrary convex sets of the form (4.1.3). In each iteration of our IRWA algorithm, a quadratic model of a relaxed problem is formed and solved to determine the next iterate. Similarly, the ADAL algorithm that we present also has as its main computational component the minimization of a convex quadratic subproblem. Both solvers can be applied in large scale settings, and both can be implemented matrix-free.

Variations of our algorithms were implemented and the performance of these implementations were tested. Our test results indicate that both types of algorithms perform similarly on many test problems. However, a test on an ℓ_1-SVM problem illustrates that in some applications the IRWA algorithms can have superior performance. While the accelerated version of both methods is the preferred implementation, we have provided global convergence and complexity results for unaccelerated variants of the algorithms. Complexity results for accelerated versions remains an open issue.

We proposed a novel dynamic penalty parameter updating strategy name DUST for matrix-free SQO for solving nonlinear optimization problems. In contemporary penalty SQO methods, the common strategy is to update the penalty parameter after a subproblem (or a sequence of them) has been solved. This may lead to inefficiencies if the parameter is slow to adapt to the problem scaling or structure. By contrast, we propose an approach to update a penalty parameter during the optimization process for each subproblem, where the goal is to produce a search direction that simultaneously predicts progress towards feasibility and optimality.

We prove that our approach yields reasonable (i.e., not excessively small) values of the penalty parameter and illustrate the behavior of our approach via numerical experiments. We first incorporate DUST in an ADAL QO solver and test it on small-scale problems, and then in a Coordinate Descent QO solver and test it on small- to medium-scale problems. Both experiments show the acceptable performance of DUSt.

We have to mention that our implementation is simply a basic SQO framework without thorough theoretical guarantees and techniques handling practical issues. Therefore, the performance of our strategy may be greatly improved after carefully designing the algorithm. We have proved the boundedness of ρ during solving a single subproblem. It is interesting to see whether ρ is still bounded near an optimal solution of the NLO when running the SQO method.

Bibliography

[1] D.H. Anderson and M.R. Osborne. Discrete, nonlinear approximation problems in polyhedral norms. *Numerische Mathematik*, 28:143–156, 1977.

[2] A.E. Beaton and J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectrographic data. *Technometrics*, 16:147–185, 1974.

[3] P. T. Boggs, J. W. Tolle, and P. Wang. On the local convergence of quasi-Newton methods for constrained optimization. *SIAM Journal on Optimal Control*, 20:161–171, 1982.

[4] E. G. Boman. *Infeasibility and negative curvature in optimization*. PhD thesis, Scientific Computing and Computational Mathematics, Stanford University, Palo Alto, CA, 1999.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[6] J. V. Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33:260–279, 1985.

[7] J. V. Burke. A sequential quadratic programming algorithm for potentially infeasible mathematical programs. *Journal of Mathematical Analysis and Applications*, 139:319–351, 1989.

[8] J. V. Burke. A robust trust region method for constrained nonlinear programming problems. *SIAM Journal on Optimization*, 2(2):325–347, 1992.

[9] J. V. Burke and S.-P. Han. A Robust Sequential Quadratic Programming Method. *Mathematical Programming*, 43:277–303, 1989.

[10] James V. Burke. Second order necessary and sufficient conditions for convex composite NDO. *Mathematical Programming*, 38:287–302, 1987.

[11] James V. Burke, Frank E. Curtis, Hao Wang, and Jiashan Wang. Iterative reweighted linear least squares for exact penalty subproblems on product sets. *SIAM Journal on Optimization*, 25(1):261–294, 2015.

[12] J.V. Burke. A sequential quadratic programming method for potentially infeasible mathematical programs. *Journal of Mathematical Analysis and Applications*, 139:319–351, 1987.

[13] R. H. Byrd, F. E. Curtis, and J. Nocedal. Infeasibility detection and SQP methods for nonlinear optimization. *SIAM Journal on Optimization*, 20(5):2281–2299, 2010.

[14] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[15] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. On the convergence of successive linear-quadratic programming algorithms. *SIAM Journal on Optimization*, 16(2):471, 2005.

[16] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1997.

[17] R. H. Byrd, G. López-Calva, and J. Nocedal. A line search exact penalty method using steering rules. *Mathematical Programming*, 2010. 0.1007/s10107-010-0408-0.

[18] R. H. Byrd, J. Nocedal, and R. A. Waltz. Knitro: an integrated package for nonlinear optimization. *Large Scale Nonlinear Optimization, Nonconvex Optimization and Its Applications*, 83:35–59, 2006.

[19] R. H. Byrd, J. Nocedal, and R. A. Waltz. Steering exact penalty methods for nonlinear programming. *Optimization Methods and Software*, 23(2):197–213, 2008.

[20] Richard H. Byrd, Frank E. Curtis, and Jorge Nocedal. An inexact sqp method for equality constrained optimization. *SIAM J. on Optimization*, 19(1):351–369, April 2008.

[21] Richard H. Byrd, Nicholas I.M. Gould, Jorge Nocedal, and Richard A. Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming*, 100(1):27–48, 2003.

[22] R. M. Chamberlain, Powell M. J. D., C. Lemaréchal, and H. C. Pedersen. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Study*, 16:1–17, 1982.

[23] C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96:161–177, 2003.

[24] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley & Sons, New York, NY, USA, 1983.

[25] A. R. Conn, N. I. M. Gould, and P. L. Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[26] D. Cores and R. A. Tapia. Perturbation lemma for the Newton method with application to the SQP Newton method. *Journal of Optimization Theory and Applications*, 97:271–280, 1998.

[27] Frank E. Curtis, Travis C. Johnson, Daniel P. Robinson, and Andreas Wchter. An inexact sequential quadratic optimization algorithm for nonlinear optimization. *SIAM Journal on Optimization*, 24(3):1041–1074, 2014.

[28] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Classics in Applied Mathematics. SIAM, Philadelphia, PA, USA, 1996.

[29] Elizabeth D. Dolan, Jorge J. Mor, and Todd S. Munson. Benchmarking optimization software with cops 3.0. In *MATHEMATICS AND COMPUTER SCIENCE DIVISION, ARGONNE NATIONAL LABORATORY*, 2004.

[30] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization.* PhD thesis, Massachusetts Institute of Technology, 1989.

[31] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *Journal of Optimization Theory and Application*, 85:265–289, 1995.

[32] F. Fletcher and E. Sainz de la Maza. Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, 43:235–256, 1989.

[33] R. Fletcher. A model algorithm for composite nondifferentiable optimization. *Mathematical Programming Study*, 17:67–76, 1982.

[34] R Fletcher. An l 1 penalty method for nonlinear constraints. In *Numerical Optimization 1984: Proceedings of the SIAM Conference on Numerical Optimization, Boulder, Colorado, June 12-14, 1984*, volume 20, page 26. Siam, 1985.

[35] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, Chichester, UK, 2 edition, 1987.

[36] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[37] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-sqp algorithm. *SIAM Journal on Optimization*, 13(1):44–59, 2002.

[38] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Brooks/Cole, 2002.

[39] M. P. Friedlander, N. I. M. Gould, and S. Leyffer. A filter active-set trust-region method. *Technical Report Preprint ANL/MCS-P1456-097. Argonne: Argonne National Laboratory*, 2007.

[40] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: an sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[41] D. Goldfarb and S. Ma. Fast multiple-splitting algorithms for convex optimization. *SIAM Journal on Optimization*, 22(2):533–556, 2012.

[42] Donald Goldfarb, Shiqian Ma, and Katya Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 141(1-2):349–382, 2013.

[43] Tom Goldstein, Brendan O'Donoghue, and Simon Setzer. Fast alternating direction optimization methods. Technical report, CAM Report 12-35, UCLA, 2012.

[44] N. I. M. Gould and D. P. Robinson. A second derivative SQP method: global convergence. *SIAM Journal of Optimization*, 20:2023–2048, 2010.

[45] N. I. M. Gould and D. P. Robinson. A second derivative SQP Method: local convergence and practical issues. *SIAM Journal of Optimization*, 20:2049–2079, 2010.

[46] N.I.M. Gould, S. Lucidi, M. Roma, and P.L. Toint. Solving the trust-region subproblem using the lanczos method. *SIAM J. Optim.*, 9:504–525, 1999.

[47] S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, 1977.

[48] Matthias Heinkenschloss and Luis N. Vicente. Analysis of inexact trust-region sqp algorithms. *SIAM J. on Optimization*, 12(2):283–302, February 2002.

[49] W. Hock and K. Schittkowski. Test examples for nonlinear programming codes. *Journal of Optimization Theory and Applications*, 30(1):127–129, 1980.

[50] Vanderbei R. J. and Shanno D. F. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

[51] T. Hastie J. Zhu, S. Rosset and R. Tibshirani. 1-norm support vector machines. *The Annual Conference on Neural Information Processing Systems*, 16:49–56, 2004.

[52] H. Jäger and E. W. Sachs. Global convergence of inexact reduced sqp methods. *Optimization Methods and Softwares*, 7:83–110, 1996.

[53] F. John. Extremum Problems with Inequalities as Side Conditions. In K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, editors, *Studies and Essays, Courant Anniversary Volume*, pages 187–204. Wiley (Interscience), New York, New York, 1948.

[54] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[55] W. Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints.* Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.

[56] H. W. Kuhn and A. W. Tucker. Nonlinear Programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Berkeley, California, 1951.

[57] Morales J. L., Nocedal J., and Wu Y. A sequential quadratic programming algorithm with an additional equality constrained phase. *IMA Journal of Numerical Analysis Advance Access*, 2011.

[58] M. Lalee, J. Nocedal, and T. D. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8:682–706, 1998.

[59] X. W. Liu and Y. Yuan. A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties. *Mathematical Programming*, 125(1):163–193, 2010.

[60] N. Maratos. *Exact penalty function algorithms for finite dimensional and control optimization problems.* PhD thesis, University of London, London, England, 1978.

[61] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[62] Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[63] D.P O'Leary. Robust regression computation using iteratively reweighted least squares. *SIAM J. Matrix Anal. Appl.*, 11:466–480, 1990.

[64] Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9:10–20, 2007.

[65] M.R. Osborne. *Finite Algorithms in Optimization and Data Analysis.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1985.

[66] P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is NP-Hard. *Journal of Global Optimization*, 1(1):15–22, 1991.

[67] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical Analysis*, Lecture Notes in Mathematics, pages 144–157. Springer, Berlin, Heidelberg, 1978.

[68] M. J. D. Powell. Variable metric methods for constrained optimization. *Mathematical Programming: The State of the Art*, pages 288–311, 1983.

[69] M.J.D. Powell. *General Algorithms for discrete nonlinear approximation calculations*. Academic Press, N.Y., 1983.

[70] S. M. Robinson. A quadratically-convergent algorithm for general nonlinear programming problems. *Mathematical Programming*, 3:145–156, 1972.

[71] S. M. Robinson. Perturbed kuhn-tucker points and rates of convergence for a class of nonlinear programming algorithms. *Mathematical Programming*, 7:1–16, 1974.

[72] R. Tyrrell Rockafellar and Roger J-B. Wets. *Variational Analysis*, volume 317 of *A Series of Comprehensive Studies in Mathematics*. Springer, 1998.

[73] R.T. Rockafellar. *Convex Analysis*. Priceton Landmarks in Mathematics. Princeton University Press, 1970.

[74] R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*, volume 317. Springer, 1998.

[75] W. Rudin. *Principles of Mathematical Analysis*, volume 3. McGraw-Hill New York, 1976.

[76] E.J. Schlossmacher. An iterative technique for absolute deviations curve fitting. *J. Am. Stat. Assoc.*, 68:857–865, 1973.

[77] J.E. Spingarn. Applications of the method of partial inverses to convex pro- gramming: decomposition. *Mathematical Programming*, 32:199–223, 1985.

[78] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming: Series A and B archive*, 106(1):25–57, 2006.

[79] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2008.

[80] R. A. Waltz, J. L. Morales, J. Nocedal, and Orban D. An interior point algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408, 2006.

[81] H. Wang and A. Banerjee. Online alternating direction method. *http://arxiv.org/abs/1306.3721*, 2013.

[82] R. B. Wilson. *A simplicial algorithm for concave programming.* PhD thesis, Graduate School of Business Administration, Harvard University, Cambridge, MA, USA, 1963.

[83] R. Wolke and H. Schwetlick. Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons. *SIAM J. Sci. Stat. Comput.*, 9:907–921, 1988.

[84] Stephen J Wright and Jorge Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[85] E. H. Zarantonello. *Projections on convex sets in Hilbert space and spectral theory.* Academic Press, New York, 1971.

Appendix

This appendix includes the small infeasible test examples, (each with only two or three variables), all of which are infeasible and have the infeasible stationary point located at the origin. Typically, algorithms for NLO require a few assumptions to guarantee nice convergence properties: *regularity*, which means the gradients of the equality constraints and active inequality constraints are linearly independent; *strict complementarity*, which requires the Lagrange multipliers for the equality constraints and active inequality constraints to be nonzero; *second-order sufficiency*, which implies the Hessian of the Lagrangian function is (sufficiently) positive definite on the null space of gradients of the equality constraints and active inequality constraints. In order to have a variety of interesting test problems, the ones we have constructed satisfy different combinations of these assumptions (when observed in the context of the infeasibility problem (3.2.2) after slack variables are added to produce a constrained problem); in total we end up with eight different combinations which are listed in the Table 1 (Y=Yes, N=No). For Example 4 and Example 6 where the regularity condition does not hold, we observe that the multipliers cannot be uniquely determined, and some of them violate strictly complementarity. Therefore, it is not possible to create an instance with the regularity condition violated and strictly complementarity always satisfied, so we indicate Y/N for strict complementarity for such cases. The formulations of these examples are given in the Appendix. Another problem, Example 9 — which is not mentioned in the table — is a feasible problem for which we find some curious results from `Filter`.

Table 1: Properties satisfied by text examples

Example	Regularity	Strictly complementarity	Second-order sufficiency
1	Y	N	Y
2	N	N	Y
3	Y	Y	Y
4	N	Y/N	Y
5	Y	Y	N
6	N	Y/N	N
7	N	N	N
8	Y	N	N

The formulations for the examples are given below, with their properties verified.

Example 1. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & -x_1^3 - x_1 \geq 0 \\ & -1 - x_1^2 - x_2^2 \geq 0. \end{aligned}$$

The only point is $(0,0)^T$ with $\lambda_1 = 0$, $\lambda_2 = 1$, $\sigma_1 = 1$, $\sigma_2 = 0$, and $r_1 = 0$, $r_2 = 1$. The first constraint is active at $(0,0)$, but the corresponding multiplier is $\lambda_1 = 0$, so the strictly complementarity condition is not satisfied. At $(0,0)^T$, the gradient of the first constraint is $(-1,0)^T$, so the regularity condition is satisfied. The matrix

$$W(x,\lambda) = -[\sum_{i\in\mathcal{A}} \lambda_i \nabla^2 g_i(x) + \sum_{i\in\mathcal{V}} \nabla^2 g_i(x)] = -\lambda_2 \nabla^2 g_2(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

is positive definite for all $d \neq 0$ and so is positive definite for all d such that $(-1,0)d = 0$. The initial point is $(10,15)$.

Example 2. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & -e^{x_1} - x_2 + 1 \geq 0 \\ & -x_1^2 + x_2 \geq 0 \\ & -x_1^2 - x_2 \geq 0 \\ & -x_2^2 - 1 \geq 0. \end{aligned}$$

$(0,0)^T$ is the only KKT point with $\lambda_1 = 0$, $\sigma_1 = 0$, $r_1 = 0$, $r_2 = r_3 = 0$, $\lambda_4 = 1$, $\sigma_4 = 0$, $r_4 = 1$, σ_2 and σ_3 can be any value in $[0,1]$ and $\lambda_2 = 1-\sigma_2$, $\lambda_3 = 1-\sigma_3$. The last constraint is not satisfied. At $(0,0)$, the gradient of active constraints are $(-1,-1)^T, (0,1)^T$, and $(0,-1)^T$. They are linearly dependent; so the regularity condition is violated. The matrix

$$\begin{aligned} W(x,\lambda) &= -\lambda_2 \nabla^2 g_2(x) - \lambda_3 \nabla^2 g_3(x) - \lambda_4 \nabla^2 g_4(x) \\ &= \lambda_2 \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \end{aligned}$$

and the set $\{d | (-1,-1)d = 0, (0,1)d = 0, (0,-1)d = 0, d \neq 0\} = \emptyset$. So the second-order sufficiency condition is satisfied. The initial point is $(20, 20)$.

Example 3. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & -x_1^2 + x_2 + 1 \geq 0 \\ & -x_1^2 - x_2^2 - 1 \geq 0. \end{aligned}$$

One can show $(0,0)^T$ is the only KKT point with $\lambda_1 = 0$, $\lambda_2 = 1$, $r_1 = 0$, $r_2 = 1$, $\sigma_1 = 1$ and $\sigma_2 = 0$. There is no active constraint. So the regularity condition naturally holds. The matrix

$$W(x,\lambda) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

is positive definite, so the second-order sufficiency condition is satisfied. The initial point is $(-20, -20)$.

Example 4. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & 0.5x_1^2 \geq 0 \\ & -x_1^2 - x_2^2 - 1 \geq 0. \end{aligned}$$

One can verify that $(0,0)^T$ is the only KKT point with λ_1 being any value in $[0,1]$, $\lambda_2 = 1$, $r_1 = 0$, $r_2 = 1$, σ_1 being any value in $[0,1]$ and $\sigma_2 = 0$. The first constraint is active and its gradient is $(0,0)^T$. The matrix

$$W(x,\lambda) = \lambda_1 \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

is positive definite (since $\lambda_1 \in [0,1]$). So the second-order sufficient condition is satisfied. The initial point is $(20,20)$.

Example 5. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & -x_1^2 + x_2 + 1 \geq 0 \\ & -x_1^4 - x_2^4 - 1 \geq 0. \end{aligned}$$

One can verify $(0,0)^T$ is the only KKT point with $\lambda_1 = 0$, $\lambda_2 = 1$, $r_1 = 0$, $r_2 = 1$, $\sigma_1 = 1$ and $\sigma_2 = 0$. There is no active constraint. So the regularity condition naturally holds. The matrix

$$W(x,\lambda) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

is not positive definite, so the second-order sufficiency condition is violated. The initial point is $(20,20)$.

Example 6. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 + x_3 \\ \text{s.t.} \quad & -x_1^4 - x_2^4 - 1 \geq 0 \\ & -x_3^4 \geq 0. \end{aligned}$$

The only KKT point is $(0,0,0)^T$ with $\lambda_1 = 1$, $\sigma_1 = 0$, $r_1 = 1$, λ_2 can be any value in $[0,1]$, $\sigma_2 = 1-\lambda_2$ and $r_2 = 0$. The second constraint is active at $(0,0,0)^T$, but the corresponding multiplier is not necessarily in $(0,1)$,so the strictly complementarity condition may not hold for some values. At $(0,0,0)^T$, the gradient of the active constraint is $(0,0,0)^T$. It is linearly dependent, so the regularity condition is violated. The matrix

$$\begin{aligned} W(x,\lambda) &= -\lambda_1 \nabla^2 g_1(x) - \lambda_2 \nabla^2 g_2(x) \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \end{aligned}$$

and the set $\{d|(0,0,0)d = 0, d \neq 0\} = \mathbb{R}^3$. So the second-order sufficiency condition is not satisfied. The initial point is $(-10, 0.5, 0.5)$.

Example 7. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 + x_3 \\ \text{s.t.} \quad & -e^{x_1} - x_2 + 1 \geq 0 \\ & -x_1^2 + x_2 \geq 0 \\ & -x_1^2 - x_2 \geq 0 \\ & -x_2^2 - x_3^4 - 1 \geq 0. \end{aligned}$$

Point $(0,0,0)^T$ is the only KKT point with $\lambda_1 = 0$, $\sigma_1 = 1$, $r_1 = 0$, $r_2 = r_3 = 0$, $\lambda_4 = 1$, $\sigma_4 = 0$, $r_4 = 1$, σ_2 and σ_3 can be any value in $[0,1]$ and $\lambda_2 = 1 - \sigma_2$, $\lambda_3 = 1 - \sigma_3$. The first constraint is active at $(0,0,0)^T$, but the corresponding multiplier is $\lambda_1 = 0$,so the strictly complementarity condition is not satisfied. At $(0,0,0)^T$, the gradients of the active constraints are $(-1,-1,0)^T$, $(0,-1,0)^T$, and $(0,-1,0)^T$. They are linearly independent, so the regularity condition is violated. The matrix

$$\begin{aligned} W(x,\lambda) &= -\lambda_2\nabla^2 g_2(x) - \lambda_3\nabla^2 g_3(x) - \lambda_4\nabla^2 g_4(x) \\ &= \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \end{aligned}$$

and the set $\{d|(-1,-1,0)d = 0, (0,1,0)d = 0, (0,-1,0)d = 0, d \neq 0\} = \{(0,0,x_3)^T|x_3 \in \mathbb{R}\}$. So the second-order sufficiency condition is not satisfied. The initial point is $(20, 20, 20)$.

Example 8. Consider

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & -x_1^3 - x_1 \geq 0 \\ & -1 - x_1^4 - x_2^4 \geq 0. \end{aligned}$$

The only KKT point is $x_1 = 0, x_2 = 0$ with $\lambda_1 = 0$, $\lambda_2 = 1$, $\sigma_1 = 1$, $\sigma_2 = 0$, and $r_1 = 0$, $r_2 = 1$. The first constraint is active, but the corresponding multiplier is $\lambda_1 = 0$, so the strictly complementarity condition is violated. At $(0,0)^T$, the gradient of the first constraint is $(-1,0)^T$, so the regularity condition is satisfied. The matrix

$$W(x,\lambda) = -[\sum_{i\in\mathcal{A}} \lambda_i \nabla^2 g_i(x) + \sum_{i\in\mathcal{V}} \nabla^2 g_i(x)] = -\lambda_2\nabla^2 g_2(x) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

is not positive definite. The initial point is $(10, 15)$.

Example 9. The problem is given by

$$\begin{aligned} \min \quad & x_1^3 + 7x_2^2 \\ \text{s.t.} \quad & 3x_2 - x_2(x_1+6)^2 - 1 \geq 0 \\ & 9(x_1+4)^2 - 5 - (x_2-3)^2 \geq 0 \\ & 2x_1 - 3(x_2+2)^2 + 104 \geq 0 \\ & -4(x_2-5)^2 - 2 - 3x_1 \geq 0. \end{aligned}$$

Biography

Hao Wang earned his Bachelor of Science and Master of Science in Applied Mathematics from Beihang University (former Beijing University of Aeronautics and Astronautics) in 2007 and 2009, respectively. In 2009, he joined the doctoral program in Industrial Engineering at Lehigh University. Whiling pursuing his degree, Hao Wang participated in research internship programs at Mitsubishi Electric Research Laboratory and ExxonMobil Corporate Strategic Research Laboratory. Hao Wang's dissertation, *Practical Enhancements in Sequential Quadratic Optimization: Infeasibility Detection, Subproblem Solution, and Penalty Parameter Updates*, was supervised by Dr. Frank E. Curtis.