Noname manuscript No.
(will be inserted by the editor)

An Sℓ_1LP-Active Set Approach for Feasibility Restoration in Power Systems

Taedong Kim · Stephen J. Wright

May 1, 2014

Abstract We consider power networks in which it is not possible to satisfy all loads at the demand nodes, due to some attack or disturbance to the network. We formulate a model, based on AC power flow equations, to restore the network to feasibility by shedding load at demand nodes, but doing so in a way that minimizes a weighted measure of the total load shed, and affects as few demand nodes as possible. This solution provides guidance to operators on how to minimize the impact of a disruptive event on those served by the grid. Optimization techniques including nonsmooth penalty functions, sequential linear programming, and active set heuristics are used to solve this model. We describe an algorithmic framework and present convergence results, including a superlinear convergence result for the case in which the solution is fully determined by its constraints, a situation that arises frequently in the power systems application.

Keywords AC power flow equations, composite nonsmooth optimization, sequential linear programming, active-set methods

1 Introduction

Consider a power grid that has experienced an unexpected event that may interfere with its ability to meet load requirements at its demand nodes. The event could be a natural contingency, an equipment failure, or a malicious

This research was supported by DOE Grant DE-SC000228, a DOE grant subcontracted through Argonne National Laboratory Award 3F-30222, and National Science Foundation Grant DMS-1216318.

T. Kim · S.J. Wright
Computer Sciences Department, 1210 W. Dayton Street, University of Wisconsin, Madison, WI 53706, USA
E-mail: tdkim@cs.wisc.edu, swright@cs.wisc.edu

attack intended to disrupt the system. The operating point needs to be modified after such an event. Since a power network often has some resistance to small perturbations, the system may still be operational after the disruption. If, however, the perturbation exceeds the tolerance of the system, the power flow problem may fail to have a solution until the configuration is adjusted, for example, by reducing loads at the demand nodes. We formulate and solve the problem of shedding loads, in the least disruptive manner possible, to restore feasible operation of the network. Our formulation is based on the nonlinear AC power-flow model, so we need to solve a nonlinear program, or minimize its equivalent nonsmooth penalty function. The measure of total load shed also serves as a metric for the severity of the disruption, which may be be useful in analyzing the vulnerability of the grid to attacks of different kinds.

1.1 Our Approach

Our approach for solving feasibility restoration problems based on AC power-flow models uses an algorithmic framework of trust-region methods for composite nonsmooth optimization (CNSO). The subproblem solved at each iteration can be posed as a linear program (LP), solved with a simplex algorithm. Although the existing power-flow literature tends to favor the use of interior-point methods, we find the LP-based approach to be appealing because of its amenability to warm starting, which often allows linearized subproblems to be solved quickly after the first few "outer" iterations (of the sequential linear programming strategy). Moreover, excellent software such as CPLEX is available for linear programming, and it can be invoked easily from such power systems modeling frameworks such as MATPOWER (Zimmerman et al 2011). By contrast, warm-starting strategies for interior-point methods have not proved to be effective in general (Yildirim and Wright 2002), except when the optimal active set does not change between outer iterations.

Because we formulate the nonlinear equality constraints in the problem using an ℓ_1-penalty term, rather than enforcing them as hard constraints, all subproblems are feasible, provided that the initial starting point for the very first subproblem is feasible. Feasible initial points can be chosen without any additional processing, since the only explicit constraints left in the subproblems after reformulation are box constraints. There is no need to complicate the algorithm by solving separately for tangential and normal steps, as is done in a number of previous approaches described below.

The first-order information used in sequential ℓ_1-linear programming (Sℓ_1LP) framework is sometimes sufficient to produce a quadratic convergence rate. When applied to a feasible system, our approach usually takes the same steps as Newton's method applied to the AC power flow equations (which are equality constraints in our formulation), and both methods require only first derivatives of the AC equations to be computed. Thus, if the power system is feasible, the convergence rate of our algorithm is quadratic, like Newton's method for nonlinear equations. We observe this fast convergence behavior too on several

infeasible networks, especially when a network can be recovered by adjusting just a few demand nodes.

In general, however, only linear convergence can be expected from subproblems constructed with first-order information. To overcome this slow rate of local convergence, we employ an active-set heuristic that can accelerate the convergence of the algorithm at the cost of second-derivative evaluations of the AC power flow formulae. In this heuristic, we estimate the optimal active set of a problem and solve the resulting equality-constrained optimization problem using a Newton-like approach. If the correct active set cannot be identified after a modest number of attempts, we revert to the Sℓ_1LP strategy.

1.2 Previous Work

There is a wide literature on applying nonlinear optimization techniques to AC power flow equations. We survey here the works most relevant to our approach from the recent literature, and then indicate how our approach differs. Most of these papers use the setting of optimal power flow (OPF) rather than feasibility, but since both these problems are closely related to the formulation we consider here, we discuss them together.

A formulation like ours for the unsolvability problem is considered in Granville et al (1996), where the objective is a weighted sum of the fractions of loads that cannot be met. A primal-dual interior-point method is proposed for the resulting nonlinear program. The algorithm is a basic interior-point method that includes few safeguards to ensure convergence.

The formulation in Barboza and Salgado (2001b) seeks to minimize the sum of squares of the loads that cannot be met, subject to the AC power flow equations being satisfied at other specified nodes, along with line and voltage magnitude limits. A primal-dual interior-point method is applied to the resulting nonlinear program. In Barboza and Salgado (2001a), the same authors consider an alternative formulation in which the rate of decrease of the load at the nodes eligible for load shedding is specified in advance, leading to a nonlinear optimization formulation whose objective consists of a single parameter. The interior-point approach is again used to solve this formulation.

Trust-region approaches have been considered by several authors, in conjunction with both interior-point and sequential quadratic programming (SQP) algorithms. All these papers have OPF as their target problem, with the AC power flow equations as constraints, along with line limits and voltage magnitude limits. In Zhou et al (2005), an SQP approach with trust regions and an ℓ_1-merit function is described. Since the primary trust-region subproblem can become infeasible for small values of the trust-region radius Δ^k, the step is separated into components that are normal and tangent to the active constraint manifold, respectively (see Nocedal and Wright (2006, Section 18.5) for a description of this technique), but it is not clear how the active set is determined for purposes of this calculation. The second-order term in each subproblem is also not specified.

Sousa and Torres (2007) describe two trust-region approaches for the OPF problem. The first approach is an SQP approach somewhat like that of Zhou et al (2005), where the tangent subproblem is solved with an interior-point method for quadratic programming. In the second approach, a nonlinear primal-dual interior-point method is applied directly to the nonlinear program. Second derivatives are used in both approaches. A journal paper by this team (Sousa et al 2011) focuses on the first approach with some modifications, chiefly, that interior-point methods for quadratic programming are used to solve both the normal and tangential subproblems. Global convergence of the approach to a local stationary point is noted, and computational results are presented on standard test sets with up to 1211 buses.

The algorithm described in Min and Shengsong (2005) is a sequential linear programming (SLP) approach with trust regions, applied to an OPF formulation, with the subproblems solved by an interior-point method for linear programming. After the SLP step is calculated, the iterate is adjusted by solving the AC power flow equations, thus ensuring that every iterate satisfies the equality constraints exactly in the nonlinear programming formulation. The authors note that the trust-region subproblem can become infeasible if the radius Δ^k is too small, so they propose a modification in which the inequality constraints are relaxed, and the amount of relaxation is penalized in the subproblem. No convergence theory is presented.

1.3 Outline

In the next section, we introduce some notation along with the problem formulation that is used as the basis of our analysis, together with optimality conditions. Section 3 describes the sequential ℓ_1 linear programming (Sℓ_1LP) framework and its global and local convergence properties. Section 4 discusses heuristics for identifying the optimal active set and estimation of Lagrange multipliers, which can be incorporated into the Sℓ_1LP framework to boost the convergence rate. Section 5 describes the application to minimal load shedding in disrupted power systems, with computational results and comparisons to other approaches described in Section 6.

2 Formulation, Overview of the Algorithm, and Notation

We begin this section by describing the problem formulation, along with its optimality conditions and its nonsmooth penalty-function equivalent. We also give a sketch of our algorithmic approach. The second subsection presents basic relevant concepts from convex analysis that are used in the convergence analysis of Section 3.

2.1 Problem Formulations and Optimality Conditions

We consider the following optimization problem:

$$\min_{x} \quad p^T x \tag{1a}$$
$$\text{subject to} \quad c(x) = 0 \tag{1b}$$
$$\underline{x} \le x \le \overline{x}, \tag{1c}$$

where p, $\underline{x}$ and $\overline{x}$ are vectors in $\mathbb{R}^n$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function with $n \ge m$. Any problem with nonlinear objectives and constraints can be formulated in this way by introducing auxiliary variables; we prefer to isolate the nonlinearity in the equality constraint to simplify the analysis and description of the method.

At a feasible point x of (1), we define the *set of active inequalities* as

$$\underline{\mathcal{A}}(x) = \{ i \,|\, x_i = \underline{x}_i,\ i = 1, 2, \ldots, n \}, \tag{2a}$$
$$\overline{\mathcal{A}}(x) = \{ i \,|\, x_i = \overline{x}_i,\ i = 1, 2, \ldots, n \}, \tag{2b}$$
$$\mathcal{A}(x) = \underline{\mathcal{A}}(x) \cup \overline{\mathcal{A}}(x), \tag{2c}$$

and the *set of inactive inequalities* as

$$\underline{\mathcal{I}}(x) = \{1, 2, \cdots, n\} \setminus \underline{\mathcal{A}}(x), \tag{3a}$$
$$\overline{\mathcal{I}}(x) = \{1, 2, \cdots, n\} \setminus \overline{\mathcal{A}}(x), \tag{3b}$$
$$\mathcal{I}(x) = \{1, 2, \cdots, n\} \setminus \mathcal{A}(x). \tag{3c}$$

For convenience, we use abbreviated notation for these sets when evaluated at an iterate x^k: $\underline{\mathcal{A}}^k := \underline{\mathcal{A}}(x^k)$, $\overline{\mathcal{A}}^k := \overline{\mathcal{A}}(x^k)$, and so on. Similarly, at a solution x^* of (1), we use $\underline{\mathcal{A}}^* := \underline{\mathcal{A}}(x^*)$, $\overline{\mathcal{A}}^* = \overline{\mathcal{A}}(x^*)$, and so on. The operator $|\cdot|$ is used to denote cardinality of a set.

The *linear independent constraint qualification (LICQ)* for (1) at x^* is:

$$\{\nabla c_i(x^*),\ i = 1, 2, \ldots, m\} \cup \{e_i,\ i \in \mathcal{A}^*\} \text{ is linearly independent}$$

where $\nabla c_i(x^*)$ is the gradient of $c_i(x)$ at x^* and e_i is the ith column of $n \times n$ identity matrix. We assume that LICQ holds for the solutions of (1).

Definition 1 We say a solution x^* of (1) is *fully determined* by the constraints if (a) $m + |\mathcal{A}^*| = n$; and (b) LICQ holds at x^*. If $m + |\mathcal{A}^*| < n$, we call the solution *underdetermined*.

The Lagrangian function $\mathcal{L}(x, \lambda, \mu, \nu)$ of (1) is defined as

$$\mathcal{L}(x, \lambda, \mu, \nu) := p^T x + \lambda^T c(x) - \mu^T (x - \underline{x}) - \nu^T (x - \overline{x}),$$

where λ, μ, and ν are the Lagrange multipliers corresponding to the equality constraints (1b), lower and bound, and upper bound of (1c), respectively. The first-order optimality conditions of (1) are given as

$$\nabla_x \mathcal{L}(x, \lambda, \mu, \nu) = 0, \tag{4a}$$
$$c(x) = 0, \tag{4b}$$
$$0 \leq x - \underline{x} \perp \mu \geq 0, \tag{4c}$$
$$0 \geq x - \overline{x} \perp \nu \leq 0, \tag{4d}$$

where $a \perp b$ indicates $a^T b = 0$ and

$$\nabla_x \mathcal{L}(x, \lambda, \mu, \nu) = p + \nabla c(x)^T \lambda - \mu - \nu.$$

By introducing the following ℓ_1-*penalty function* $\phi(x)$ with a penalty parameter $\omega > 0$:

$$\phi(x) := p^T x + \omega \left\| c(x) \right\|_1, \tag{5}$$

we can reformulate problem (1) as a CNSO problem with box constraints:

$$\min_x \quad \phi(x) \quad \text{subject to} \quad \underline{x} \leq x \leq \overline{x}. \tag{6}$$

It is well known that if the penalty parameter ω is sufficiently large, under reasonable conditions, a local solution of (6) is a local minimizer of (1) (Nocedal and Wright 2006, Theorem 17.3). In our algorithm, the CNSO problem is solved using a trust-region framework, for a particular choice of parameter ω. At each iteration, we define a linearized model of the objective in (6) and solve an LP subproblem containing the constraints in (6) along with a trust region. Following Fletcher (1987), we refer to this approach as sequential ℓ_1 linear programming (Sℓ_1LP). The trust region is adjusted so that the step obtained from the LP subproblem gives a "sufficient decrease" in the objective ϕ at each iteration, guaranteeing global convergence. We will show that for fully determined solutions, the Sℓ_1LP algorithm converges quadratically under certain conditions.

If the solution of a problem is *underdetermined*, we do not expect the fast convergence of Sℓ_1LP, so we enhance the basic strategy with active-set heuristics that use second-order information, to recover rapid local convergence.

Let x^* be a (local) optimal solution of (1), satisfying (4). Given the optimal active and inactive sets at x^* (see (2) and (3)), we can rewrite the conditions (4) as follows:

$$\begin{aligned}
\nabla_x \mathcal{L}(x, \lambda, \mu, \nu) &= 0 \\
c(x) &= 0 \\
x_i - \underline{x}_i = 0, \quad \mu_i \geq 0 \ &\text{for} \ i \in \underline{\mathcal{A}}^* \\
x_i - \overline{x}_i = 0, \quad \nu_i \leq 0 \ &\text{for} \ i \in \overline{\mathcal{A}}^* \\
\underline{x}_i < x_i < \overline{x}_i \ &\text{for} \ i \in \mathcal{I}^* \\
\mu_i = 0 \ &\text{for} \ i \in \underline{\mathcal{I}}^* \\
\nu_i = 0 \ &\text{for} \ i \in \overline{\mathcal{I}}^*
\end{aligned}$$

By gathering the equality constraints in this system, we obtain

$$\begin{bmatrix} \nabla_x \mathcal{L}(x, \lambda, \mu, \nu) \\ c(x) \\ (x - \underline{x})_{\underline{\mathcal{A}}^*} \\ (x - \overline{x})_{\overline{\mathcal{A}}^*} \\ \mu_{\underline{\mathcal{I}}^*} \\ \nu_{\overline{\mathcal{I}}^*} \end{bmatrix} = 0, \tag{7}$$

whose solution $(x^*, \lambda^*, \mu^*, \nu^*)$ also satisfies the following inequalities:

$$\begin{aligned} \underline{x}_i < x_i < \overline{x}_i \ \text{ for } \ i \in \mathcal{I}^*, \\ \mu_i \geq 0 \ \text{ for } \ i \in \underline{\mathcal{A}}^*, \\ \nu_i \leq 0 \ \text{ for } \ i \in \overline{\mathcal{A}}^*. \end{aligned} \tag{8}$$

If a starting point close enough to the optimum can be identified, and if the optimal active sets are known, it may be possible to find a KKT point (4) by applying Newton's method for nonlinear equations to (7), then checking that the solution so obtained satisfies (8). To initiate this process, we need reliable ways to identify the optimal active set, and to estimate the values of the Lagrange multipliers. Under certain conditions, both active sets and the optimal Lagrange multipliers can be estimated from the duals of the LP subproblems at the previous Sℓ_1LP iteration. If after making a modest number of guesses of the optimal active set, the strategy does not appear to be converging rapidly, we return to the Sℓ_1LP strategy.

2.2 Convex Analysis Terminology and Notation

Given a closed convex set $C \subset \mathbb{R}^n$ we denote the *dual cone* by C^*, where

$$C^* := \{p \,|\, p^T t \geq 0 \text{ for all } t \in C\}. \tag{9}$$

The *polar cone* is denoted by C°, where

$$C^\circ := \{p \,|\, p^T t \leq 0 \text{ for all } t \in C\}.$$

Note that $C^\circ = -C^*$.

The set $\Omega \subset \mathbb{R}^n$ is a *polyhedral convex set* if there is a finite collection of vectors $b_i \in \mathbb{R}^n$ and scalars γ_i, $i = 1, 2, \ldots, K$, such that

$$\Omega = \{x \,|\, b_i^T x \leq \gamma_i, \ i = 1, 2, \ldots, K\}$$

(see (Rockafellar 1970, p. 170)). The active set $\mathcal{A}(x)$ at a given $x \in \Omega$ is

$$\mathcal{A}(x) := \{i = 1, 2, \ldots, K \,|\, b_i^T x = \gamma_i\}.$$

The normal cone $N_\Omega(x)$ to Ω at x is defined as

$$\begin{aligned} N_\Omega(x) :=& \{y \,|\, y^T(x' - x) \leq 0, \text{ for all } x' \in \Omega\} \\ =& \left\{ 0 + \sum_{i \in \mathcal{A}(x)} \lambda_i b_i, \,|\, \lambda_i \geq 0 \text{ for all } i \in \mathcal{A}(x) \right\}. \end{aligned} \tag{10}$$

The set of feasible directions for Ω at $x \in \Omega$ is defined as

$$\begin{aligned} \mathcal{F}_\Omega(x) :=& \{d \,|\, x + \alpha d \in \Omega \text{ for all } \alpha \text{ sufficiently small and positive}\} \\ =& \{d \,|\, b_i^T d \leq 0 \text{ for all } i \in \mathcal{A}(x)\}. \end{aligned} \tag{11}$$

It is easy to show (using a theorem of the alternative) that $N_\Omega(x) = \mathcal{F}_\Omega(x)^\circ$.

The subdifferential $\partial h(c)$ of a convex function $H : \mathbb{R}^m \to \mathbb{R}$ is defined as

$$\partial h(c) := \{v \,|\, h(c) + v^T(c' - c) \leq h(c') \text{ for all } c' \in \operatorname{dom} h\}.$$

We can obtain an explicit form of the subdifferential when h is a polyhedral convex function of the form

$$h(c) := \max_{j=1,2,\ldots,M} h_j^T c + \beta_j, \tag{12}$$

for some $h_j \in \mathbb{R}^m$ and $\beta_j \in \mathbb{R}$, $j = 1, 2, \ldots, M$. Defining the set of active hyperplanes

$$\mathcal{H}(c) := \{j = 1, 2, \ldots, M \,|\, h(c) = h_j^T c + \beta_j\},$$

we have

$$\partial h(c) = \left\{ v = \sum_{j \in \mathcal{H}(c)} \lambda_j h_j \,|\, \lambda_j \geq 0, \sum_{j \in \mathcal{H}(c)} \lambda_j = 1 \right\}.$$

2.3 Other Notation

We use $\mathbf{1}$ to denote the vector of ones, and e_i to denote the ith column of the identity matrix. Given a set $\mathcal{A} \subset \{1, 2, \ldots, n\}$, we use $I_{\mathcal{A},\cdot}$ to denote the matrix whose rows are the rows of the $n \times n$ identity matrix corresponding to the entries in $\mathcal{A}$.

3 Sequential ℓ_1 Linear Programming (Sℓ_1LP)

We begin this section by motivating and describing the Sℓ_1LP approach for problem (6). Global convergence properties are discussed in Subsection 3.2, and local convergence properties are the focus of Subsection 3.3.

3.1 Algorithm Description

To solve (6) using the Sℓ_1LP framework, the LP subproblems with an ℓ_∞-trust-region are defined as follows, at iterate x^k:

$$\min_d \quad m^k(d) \tag{13a}$$
$$\text{subject to} \quad \underline{x} \le x^k + d \le \overline{x} \tag{13b}$$
$$\|d\|_\infty \le \Delta^k, \tag{13c}$$

where $\Delta^k > 0$ is a trust region radius and the linearized *model function* $m^k(d)$ is defined as follows:

$$m^k(d) := p^T(x^k + d) + \omega \left\|c(x^k) + \nabla c(x^k)d\right\|_1 .$$

The function $m^k(d)$ is an approximation to the ℓ_1-penalty function $\phi(x^k + d)$ that we "trust" to be a good approximation in the region $\|d\| \le \Delta^k$. If a solution d^k of (13) yields a "sufficient decrease" in ϕ, the new iterate x^{k+1} is define by $x^{k+1} = x^k + d^k$. Otherwise, we reject the step d^k, define the next iterate to be $x^{k+1} = x^k$, reduce the trust-region radius, and proceed to the next iteration.

The quality of step d^k is determined by means of the actual reduction $\Delta\phi^k(d)$, the expected reduction $\Delta m^k(d)$, and the ratio $\rho^k(d)$ between these quantities, defined as follows:

$$\Delta\phi^k(d) = \phi(x^k) - \phi(x^k + d) \quad \text{(Actual reduction)} \tag{14a}$$
$$\Delta m^k(d) = m^k(0) - m^k(d) \quad \text{(Expected reduction)} \tag{14b}$$
$$\rho^k(d) = \frac{\Delta\phi^k(d)}{\Delta m^k(d)} \quad \text{(Agreement ratio).} \tag{14c}$$

The expected reduction (14b) is always nonnegative when x^k is feasible for (6), since $d = 0$ is a feasible point of (13). If $\rho^k(d^k)$ exceeds a positive threshold value $\underline{\rho}$, the step improves the objective ϕ significantly, so we can accept it. If $\rho^k(d^k)$ is close to 1, m^k and ϕ are probably in good agreement over the current trust region, so we increase the trust-region radius for the next iteration. If $\rho^k(d^k)$ falls below another small positive threshold $\underline{\eta}$, we deem the agreement between m^k and ϕ to be poor, and we reduce the trust-region for the next iteration. The kth iteration of Sℓ_1LP is specified in Algorithm 1.

We reformulate the subproblem (13) as a true linear program by introducing an auxiliary variables α to replace the ℓ_1-norm, and replacing the ℓ_∞-norm by bounds, as follows:

$$\min_{d,\alpha} \quad p^T d + \omega \sum_{i=1}^{m} \alpha_i \tag{15a}$$
$$\text{subject to} \quad -\alpha \le c(x^k) + \nabla c(x^k)d \le \alpha \tag{15b}$$
$$\max\left\{\underline{x} - x^k, -\Delta^k\right\} \le d \le \min\left\{\overline{x} - x^k, \Delta^k\right\}, \tag{15c}$$

Algorithm 1 Sequential ℓ_1 Linear Programming - iteration k

Require:
Current iterate x^k and trust-region radius Δ^k.
Upper bound on trust-region radius $\overline{\Delta}$.
Parameters $0 < \underline{\rho} \leq \underline{\eta}, \overline{\eta}, c_2 < 1 < c_1$ where $\underline{\rho} < \underline{\eta}$.
Ensure:
Next iterate x^{k+1} and trust-region radius Δ^{k+1}

Solve LP subproblem (15) to obtain step d^k.
if $\rho^k(d^k) > \underline{\rho}$ **then** ▶ Step d^k is good.
  $x^{k+1} \leftarrow x^k + d^k$
else ▶ Step d^k is poor.
  $x^{k+1} \leftarrow x^k$
end if
if $\left|1 - \rho^k(d^k)\right| < \overline{\eta}$ **then** ▶ Good agreement.
  $\Delta^{k+1} \leftarrow \min\left\{c_1\Delta^k, \overline{\Delta}\right\}$
else if $\rho^k(d^k) < \underline{\eta}$ **then** ▶ Poor agreement.
  $\Delta^{k+1} \leftarrow c_2\Delta^k$
else
  $\Delta^{k+1} \leftarrow \Delta^k$
end if

where max and min are applied element-wise. We have already mentioned that the subproblem is always feasible no matter how small the trust-region radius Δ^k is, provided x^k is feasible for the simple constraints in (6). Due to the constraint (15c), the next iterate $x^{k+1} = x^k + d^k$ or $x^{k+1} = x^k$ also satisfies the bounds in inequality constraints in (6). Thus feasibility can be guaranteed for all iterates x^k, provided that the initial iterate x^0 is feasible.

The algorithm terminates if any of the following conditions are satisfied.

(a) The objective value is equal to the known best possible value of the problem. (For example, in our application of Section 5, we can terminate if the load shedding is reduced to zero.)
(b) For some $\epsilon > 0$,

$$\frac{\left|\phi(x^k) - m^k(d^k)\right|}{\Delta^k} \leq \epsilon. \tag{16}$$

(c) The trust region radius drops below a lower bound $\underline{\Delta}$.
(d) The iteration counter reaches a maximum limit.

Condition (c) and (d) indicate unsuccessful termination. If (b) is satisfied, the ratio of possible improvement to trust region size is small, indicating that the objective is in a flat region, probably near a solution.

The Sℓ_1LP approach has several advantages. First, it does not require second derivative information, which may be expensive to evaluate. Second, as we discuss later, rapid convergence can still be achieved if the limit is a fully determined solution. Third, the subproblem is a linear program, so a simplex code can exploit warm start information from the previous Sℓ_1LP iteration.

3.2 Global Convergence Properties

In discussing the global convergence properties of Sℓ_1LP we consider a slight generalization of the formulation (6), which we define as follows:

$$\min_x \quad \phi(x) := p^T x + h(c(x)) \quad \text{subject to} \quad x \in \Omega, \tag{17}$$

where $\Omega \subseteq \mathbb{R}^n$ is a polyhedral convex set and $h : \mathbb{R}^m \to \mathbb{R}$ is a polyhedral convex function (12). The linear subproblem at iteration k for (17) is

$$\begin{aligned} \min_d \quad & m^k(d) := p^T(x^k + d) + h(c(x^k) + \nabla c(x^k)d) \\ \text{subject to} \quad & x^k + d \in \Omega, \\ & \|d\|_\infty \leq \Delta^k. \end{aligned} \tag{18}$$

We recover (6) and (13), respectively, by setting $h(x) = \omega \|x\|_1$ and $\Omega = \{x \in \mathbb{R}^n \,|\, \underline{x} \leq x \leq \overline{x}\}$ in (17) and (18).

The following technical results are useful in proving global convergence. (We omit the proof of the first result, which is elementary.)

Lemma 1 *Let C be a polyhedral convex set in $\mathbb{R}^n$. Then the dual cone C^* of C (see (9)) is a closed convex polyhedral cone.*

Lemma 2 *Let C be a nonempty closed convex polyhedron and D a nonempty closed convex polyhedral cone in $\mathbb{R}^n$. If for each $d \in D$ there exists $c \in C$ such that $c^T d \geq 0$, then there exists $c' \in C$ such that $(c')^T d \geq 0$ for all $d \in D$.*

Proof Note that $0 \in D$, since D is a closed convex cone. If $0 \in C$, we can choose $c' = 0$ and we are done. If $0 \notin C$, assume for a contradiction that for each $d \in D$ there exists $c \in C$ such that $c^T d \geq 0$ but there is no $c' \in C$ such that $(c')^T d \geq 0$ for *all* $d \in D$. Let D^* be the dual cone of D. Then D^* is a closed convex polyhedral cone by Lemma 1, and we have $C \cap D^* = \emptyset$. By the separating theorem for polyhedra (see for example Vanderbei (2007, Theorem 10.4)), there exists a vector p such that $p^T c < 0$ for all $c \in C$ and $p^T d^* \geq 0$ for all $d^* \in D^*$. From the latter, we have $p \in D^{**} = \text{cl conv cone } D = D$. Thus we have identified $p \in D$ such that $p^T c < 0$ for all $c \in C$. This contradicts the assumption mentioned before. Thus there exists $c' \in C$ such that $(c')^T d \geq 0$ for all $d \in D$. □

Now we are ready to show the global convergence of the Sℓ_1LP algorithm. For this purpose we recall the definition (10) of the normal cone $N_\Omega(x)$ to Ω at a point $x \in \Omega$. We also need the first-order optimality condition for (17) at a point x^*, which is that there exists $\lambda^* \in \partial h(c(x^*))$ such that

$$0 \in p + \nabla c(x^*)^T \lambda^* + N_\Omega(x^*). \tag{19}$$

Theorem 3 *Assume that $\phi(x)$ is bounded below and the sequence $\{x^k\}$ generated by Algorithm 1 is bounded. Also assume that $\partial h(c)$ is bounded for all c. Then there is a subsequence S of $\{x^k\}$ with a limit point x^∞ that satisfies the first-order optimality conditions (19).*

We note that the assumptions for Theorem 3 hold for our problem since $h(c) = \omega \|c\|_1$ and so $\partial h(x) \subset \otimes_{i=1}^{m}[-\omega, \omega]$, while the objective $\phi(x)$ is bounded below by zero. Versions of this result without the constraint $x \in \Omega$ appear in Fletcher (1987, Theorem 14.5.1) and Fletcher and Sainz de la Maza (1989, Theorem 2.1). Our proof follows these, with modifications to handle the presence of the feasible set Ω.

Proof By taking a further subsequence S is necessary, we can assume that one of the following two cases occurs for indices $k \in S$:

(a) $\rho^k < \underline{\eta}$ and $\Delta^{k+1} \to 0$. Thus $\|d^k\| \to 0$.
(b) $\rho^k > \underline{\eta}$ and $\inf \Delta^k > 0$.

We consider first case (a). Suppose for contradiction that there exists a nonzero vector $s \in \mathcal{F}_\Omega(x^\infty)$ and a constant $\beta > 0$ such that

$$\max_{\lambda \in \partial h(c(x^\infty))} \frac{1}{\|s\|} s^T (p + \nabla c(x^\infty)^T \lambda) = -\beta, \quad \beta > 0. \tag{20}$$

where the closed convex cone of feasible directions $\mathcal{F}_\Omega(x)$ is defined in (11). Because Ω is polyhedral convex, we have $N_\Omega(x) \subseteq N_\Omega(x^\infty)$ for all x close enough to x^∞, by outer-semicontinuity of $N_\Omega(x)$ (Rockafellar and Wets 1998, Proposition 6.6). (This fact is a consequence of $\mathcal{A}(x) \subset \mathcal{A}(x^\infty)$ for x close enough to x^∞.) We thus have $\mathcal{F}_\Omega(x^\infty) \subseteq \mathcal{F}_\Omega(x)$, since $\mathcal{F}_\Omega(x) = N_\Omega(x)^\circ$. Since $s \in \mathcal{F}_\Omega(x^\infty)$ and $\lim_{k \in S} x^k = x^\infty$, we have $s \in \mathcal{F}_\Omega(x^k)$ for k sufficiently large. Furthermore, it is easy to show from the definitions associated with polyhedral convex Ω at the end of Section 2 that $x^k + \|d^k\| (s/\|s\|) \in \Omega$, since $\|d^k\| \to 0$ and $x^k \to x^\infty$. Since $c(x)$ is continuous and ∂h is bounded, we know from Taylor's Theorem that

$$\begin{aligned} \phi(x^k + d) &= p^T (x^k + d) + h(c(x^k + d)) \\ &= p^T (x^k + d) + h\left(c(x^k) + \nabla c(x^k) d + o(\|d\|)\right) \\ &= p^T (x^k + d) + h(c(x^k) + \nabla c(x^k) d) + o(\|d\|) \\ &= m^k(d) + o(\|d\|). \end{aligned} \tag{21}$$

Then for sufficiently large k, we have

$$\begin{aligned} \Delta m^k(d^k) &= \phi(x^k) - m^k(d^k) \\ &\geq \phi(x^k) - m^k\left(\|d^k\| \frac{s}{\|s\|}\right) \\ &= \phi(x^k) - \phi\left(x^k + \|d^k\| \frac{s}{\|s\|}\right) + o\left(\|d^k\|\right) \\ &\geq \beta \|d^k\| + o\left(\|d^k\|\right). \end{aligned}$$

The first inequality follows from the fact that d^k solves the subproblem (18) whereas $s(\|d^k\|/\|s\|)$ is another feasible point for this subproblem. The second

equality follows from (21), and the final inequality follows from Fletcher (1987, Corollary to Lemma 14.5.1) and (20). Also we know that

$$\Delta\phi^k(d^k) = \Delta m^k(d^k) + o(\|d^k\|),$$

so that

$$\rho^k = \frac{\Delta\phi^k(d^k)}{\Delta m^k(d^k)} = 1 + o(1).$$

This contradicts the assumption that $\rho^k < \underline{\eta}$. Therefore there is no such vector $s \in \mathcal{F}_\Omega(x^\infty)$ for which (20) holds, so that for all $s \in \mathcal{F}_\Omega(x^\infty)$, we have

$$\max_{\lambda \in \partial h(c(x^\infty))} \quad \frac{1}{\|s\|} s^T(p + \nabla c(x^\infty)^T \lambda) \geq 0.$$

We can now set $D = \mathcal{F}_\Omega(x^\infty)$ and $C = \partial h(c(x^\infty))$ in Lemma 2 to conclude that there exists $\lambda^* \in \partial h(c(x^\infty))$ such that $s^T(p + \nabla c(x^\infty)^T \lambda^*) \geq 0$ for all $s \in \mathcal{F}_\Omega(x^\infty)$. We therefore have the desired result that $0 \in p + \nabla c(x^\infty)^T \lambda^* + N_\Omega(x^\infty)$.

In case (*b*), we know that $\Delta\phi^k(d^k) \to 0$ from

$$\phi(x^0) - \phi(x^\infty) \geq \sum_{k \in S} \Delta\phi^k(d^k).$$

Since $\rho^k \geq \underline{\eta}$, we also have $\Delta m^k(d^k) \to 0$. Now let d^∞ be a solution of the subproblem (18) with $x^k = x^\infty$ and $\Delta^\infty < \tilde{\Delta} := \inf_{k \in S} \Delta^k$. Also define $\tilde{x} = x^\infty + d^\infty$. Then

$$\begin{aligned} \|\tilde{x} - x^k\| &\leq \|\tilde{x} - x^\infty\| + \|x^\infty - x^k\| \\ &= \|d^\infty\| + o(1) \\ &\leq \Delta^\infty + o(1) \\ &\leq \Delta^k, \end{aligned}$$

for sufficiently large $k \in S$. Thus $\tilde{x} - x^k$ is feasible for $\Delta m^k(d)$ and

$$m^k(\tilde{x} - x^k) \geq m^k(d^k) = \phi(x^k) - \Delta m^k(d^k).$$

By taking limits of both sides, we have

$$m^\infty(d^\infty) \geq \phi(x^\infty) = m^\infty(0).$$

Therefore, 0 is also a solution of (18) (with $x^k = x^\infty$ and $\Delta^k = \Delta^\infty$), in particular, the trust-region constraint is inactive. From the optimality conditions of (18), there exists $\lambda^* \in \partial h(c(x^\infty))$ such that

$$0 \in p + \nabla c(x^\infty)^T \lambda^* + N_\Omega(x^\infty).$$

Thus x^∞ is a KKT point of (17). □

3.3 Fast Local Convergence in the Fully Determined Case

As discussed in Section 2, we cannot expect the Sℓ_1LP algorithm to have a local convergence rate faster than linear in general, because it uses only first-order information about the function $c(x)$. However, when the solution x^* is fully determined (see Definition 1), and when certain other conditions hold, the algorithm converges locally at a quadratic rate. In this subsection, we analyze this phenomenon, which is commonly observed in our application of Section 5.

First-order optimality conditions for problem (6) are as follows (cf. (4)):

$$p + \omega \nabla c(x)^T \lambda - \mu - \nu = 0, \tag{22a}$$

$$\lambda \in \partial \left\| c(x) \right\|_1, \tag{22b}$$

$$0 \le x - \underline{x} \perp \mu \ge 0, \tag{22c}$$

$$0 \ge x - \overline{x} \perp \nu \le 0, \tag{22d}$$

where λ, μ and ν are the dual variables. We say that *strict complementarity* holds at a primal-dual solution $(x^*, \lambda^*, \mu^*, \nu^*)$ of (22) if the following conditions are satisfied:

(*a*) $\lambda_j^* \in (-1, 1)$ if $c_j(x^*) = 0$ for $j \in \{1, 2, \ldots, m\}$,
(*b*) $\mu_i^* > 0$ if $(x^* - \underline{x})_i = 0$ for $i \in \{1, 2, \ldots, n\}$, that is, for $i \in \underline{\mathcal{A}}^*$,
(*c*) $\nu_i^* < 0$ if $(x^* - \overline{x})_i = 0$ for $i \in \{1, 2, \ldots, n\}$, that is, for $i \in \overline{\mathcal{A}}^*$.

In other words, strict complementarity requires existence of $\gamma > 0$ such that

$$\left|\lambda_j^*\right| \le 1 - \gamma \quad \text{for } j \in \{1, 2, \ldots, m\}, \tag{23a}$$

$$\mu_i^* \ge \gamma \qquad \text{for } i \in \underline{\mathcal{A}}^*, \tag{23b}$$

$$\nu_i^* \le -\gamma \qquad \text{for } i \in \overline{\mathcal{A}}^*. \tag{23c}$$

Guided by the optimality conditions (22), we have the following result concerning solution d^k of the LP subproblem (15).

Lemma 4 *Suppose that $(x^*, \lambda^*, \mu^*, \nu^*)$ satisfies (22), that x^* is fully determined, and that strict complementarity holds at $(x^*, \lambda^*, \mu^*, \nu^*)$. Then if x^k close to x^*, we have that for all Δ^k sufficiently large, the subproblem (15) has a solution (d^k, α^k) such that $\alpha^k = 0$, $(x^k + d^k)_i = \underline{x}_i$ for $i \in \underline{\mathcal{A}}^*$, $(x^k + d^k)_i = \overline{x}_i$ for $i \in \overline{\mathcal{A}}^*$ and $\left\| x^k + d^k - x^* \right\| = O(\left\| x^k - x^* \right\|^2)$.*

Proof We write the first-order necessary conditions of the subproblem (13) *without* the trust-region constraints as follows:

$$p + \omega \nabla c(x^k)^T \lambda - \mu - \nu = 0 \tag{24a}$$

$$\lambda \in \partial \left\| c(x^k) + \nabla c(x^k) d \right\|_1 \tag{24b}$$

$$0 \le x^k + d - \underline{x} \perp \mu \ge 0 \tag{24c}$$

$$0 \ge x^k + d - \overline{x} \perp \nu \le 0. \tag{24d}$$

We will construct a solution $(d^k, \lambda^k, \mu^k, \nu^k)$ to these conditions and then show that $d^k = x^* - x^k + O(\|x^k - x^*\|^2)$ and thus $\|d^k\| \le \Delta^k$ when $\|x^k - x^*\|$ is sufficiently small and Δ^k is sufficiently large. In addition, this d^k satisfies the other conditions mentioned in the theorem, and together with $\alpha^k = 0$, it solves (15).

In constructing our solution $(d^k, \lambda^k, \mu^k, \nu^k)$ to (24), we define

$$\mu_i^k = 0 \ \text{ for } \ i \notin \underline{\mathcal{A}}^* \tag{25a}$$

$$\nu_i^k = 0 \ \text{ for } \ i \notin \overline{\mathcal{A}}^*. \tag{25b}$$

and

$$d_i^k = \underline{x}_i - x_i^k = x_i^* - x_i^k \ \text{ for } \ i \in \underline{\mathcal{A}}^* \tag{26a}$$

$$d_i^k = \overline{x}_i - x_i^k = x_i^* - x_i^k \ \text{ for } \ i \in \overline{\mathcal{A}}^*. \tag{26b}$$

We further require d^k to satisfy

$$c(x^*) = 0 = c(x^k) + \nabla c(x^k) d^k. \tag{27}$$

The remaining components of $(d^k, \lambda^k, \mu^k, \nu^k)$ are required to satisfy the following linear system:

$$\begin{bmatrix} \nabla c(x^k)^T & I_{\underline{\mathcal{A}}^*}^T & I_{\overline{\mathcal{A}}^*}^T \end{bmatrix} \begin{bmatrix} \omega\lambda^k \\ -\mu_{\underline{\mathcal{A}}^*}^k \\ -\nu_{\overline{\mathcal{A}}^*}^k \end{bmatrix} = -p, \tag{28}$$

which is essentially (24a) with the substitution (25). By comparing (28) with (22a), we have

$$\begin{aligned} \begin{bmatrix} \nabla c(x^k)^T & I_{\underline{\mathcal{A}}^*}^T & I_{\overline{\mathcal{A}}^*}^T \end{bmatrix} \begin{bmatrix} \omega\lambda^k \\ -\mu_{\underline{\mathcal{A}}^*}^k \\ -\nu_{\overline{\mathcal{A}}^*}^k \end{bmatrix} &= \begin{bmatrix} \nabla c(x^*)^T & I_{\underline{\mathcal{A}}^*}^T & I_{\overline{\mathcal{A}}^*}^T \end{bmatrix} \begin{bmatrix} \omega\lambda^* \\ -\mu_{\underline{\mathcal{A}}^*}^* \\ -\nu_{\overline{\mathcal{A}}^*}^* \end{bmatrix} \\ &= \begin{bmatrix} \nabla c(x^k)^T & I_{\underline{\mathcal{A}}^*}^T & I_{\overline{\mathcal{A}}^*}^T \end{bmatrix} \begin{bmatrix} \omega\lambda^* \\ -\mu_{\underline{\mathcal{A}}^*}^* \\ -\nu_{\overline{\mathcal{A}}^*}^* \end{bmatrix} + O(\|x^k - x^*\|). \end{aligned}$$

Since LICQ holds at x^* and $m + |\mathcal{A}^*| = n$, $\begin{bmatrix} \nabla c(x^*) \\ I_{\mathcal{A}^*} \end{bmatrix}$ is a nonsingular square matrix. Then by continuity of ∇c, the matrix $\begin{bmatrix} \nabla c(x^k) \\ I_{\mathcal{A}^*} \end{bmatrix}$ is uniformly nonsingular for $\|x^k - x^*\|$ sufficiently small, and we have

$$\begin{bmatrix} \omega\lambda^k \\ -\mu_{\underline{\mathcal{A}}^*}^k \\ -\nu_{\overline{\mathcal{A}}^*}^k \end{bmatrix} = \begin{bmatrix} \omega\lambda^* \\ -\mu_{\underline{\mathcal{A}}^*}^* \\ -\nu_{\overline{\mathcal{A}}^*}^* \end{bmatrix} + O(\|x^k - x^*\|).$$

It follows from this relation and the strict complementarity assumption on $(x^*, \lambda^*, \mu^*, \nu^*)$ that $\mu^k_{\underline{\mathcal{A}}^*} > 0$, $\nu^k_{\overline{\mathcal{A}}^*} < 0$, and $\lambda^k \in (-1,1)^m$ for $\|x^k - x^*\|$ sufficiently small. Because of (27), these values are feasible for (24).

Note from (27) and Taylor's theorem that

$$\nabla c(x^k)d^k = c(x^*) - c(x^k) = \nabla c(x^k)(x^* - x^k) + O(\|x^* - x^k\|^2).$$

By combining this expression with (26), we obtain

$$\begin{bmatrix} \nabla c(x^k) \\ I_{\mathcal{A}^*} \end{bmatrix} d^k = \begin{bmatrix} \nabla c(x^k) \\ I_{\mathcal{A}^*} \end{bmatrix} (x^* - x^k) + O(\|x^* - x^k\|^2).$$

By invertibility of the coefficient matrix (discussed above), we have

$$d^k = (x^* - x^k) + O(\|x^* - x^k\|^2). \tag{29}$$

This estimate implies that the bounds inactive at x^* will also be inactive at the solution of the subproblem (15), for x^k close enough to x^*.

At this point, we have found $(d^k, \lambda^k, \mu^k, \nu^k)$ satisfying (24), and that d^k satisfies the other properties claimed in the theorem. Thus $(d, \alpha) = (d^k, 0)$ solves (15) provided that Δ^k is large enough to make the trust region inactive, that is, $\Delta^k > \|d^k\|$, where d^k is defined as above. □

Lemma 4 says that under the given conditions, there exists a solution d^k of the linearized subproblem that is a quadratic step to a solution of the problem (6). We show now that d^k is *unique* solution of the linearized subproblem, under the given conditions.

Lemma 5 *Under the conditions of Lemma 4, the vector d^k described in this result, together with $\alpha^k = 0$, is the unique solution of the linearized subproblem (15).*

Proof Assume that the conditions in Lemma 4 hold and let $(d^k, \lambda^k, \mu^k, \nu^k)$ satisfy (24). From (24a), we have for all d that

$$(p + \omega \nabla c(x^k)^T \lambda^k)^T (d - d^k) = (\mu^k + \nu^k)^T (d - d^k).$$

Since $m^k(d)$ is a convex function, we have

$$\begin{aligned} m^k(d) &\geq m^k(d^k) + (p + \omega \nabla c(x^k)^T \lambda^k)^T (d - d^k) \\ &= m^k(d^k) + (\mu^k + \nu^k)^T (d - d^k). \end{aligned}$$

We also know that

$$(\mu^k + \nu^k)_i \begin{cases} > 0 & \text{if } i \in \underline{\mathcal{A}}^*, \\ < 0 & \text{if } i \in \overline{\mathcal{A}}^*, \\ = 0 & \text{otherwise}. \end{cases} \tag{30}$$

Now assume that d' is a solution of (13). If $d'_{\mathcal{A}^*} = d^k_{\mathcal{A}^*}$, then both of d' and d^k satisfy the following equation:

$$\begin{bmatrix} \nabla c(x^k) \\ I_{\mathcal{A}^*} \end{bmatrix} d = \begin{bmatrix} c(x^*) - c(x^k) \\ (x^* - x^k)_{\mathcal{A}^*} \end{bmatrix},$$

which implies that $d' = d^k$, by nonsingularity of the coefficient matrix. If $d'_{\mathcal{A}^*} \neq d^k_{\mathcal{A}^*}$, we since d^k_i achieves its lower bound for $i \in \underline{\mathcal{A}}^*$ and its upper bound for $i \in \overline{\mathcal{A}}^*$ that

$$(d' - d^k)_{\underline{\mathcal{A}}^*} \geq 0, \quad (d' - d^k)_{\overline{\mathcal{A}}^*} \leq 0, \quad (d' - d^k)_{\mathcal{A}^*} \neq 0.$$

By (30), we have $(\mu^k + \nu^k)^T (d' - d^k) > 0$, which implies

$$m^k(d') \geq m^k(d^k) + (\mu^k + \nu^k)^T (d' - d^k) > m^k(d^k).$$

Thus d' cannot be a solution of (13). □

We now show quadratic local convergence of Algorithm 1 applied to (6), under the assumptions of this section.

Theorem 6 *If the conditions in Lemma 4 are satisfied and the initial guess x^0 is close enough to x^* and Δ^0 is sufficiently large, then the sequence $\{x^k\}$ generated by the Algorithm 1 converges to x^**

Proof The main part of the proof is to show that the step d^k defined in Lemma 4 is accepted by the algorithm, with a value of $\rho^k(d^k)$ close to 1, so that the trust region radius is not decreased, and hence remains inactive at the next iteration.

For d^k defined in Lemma 4, we have the definition (14c) that

$$\begin{aligned}
\rho^k(d^k) &= \frac{\phi(x^k) - \phi(x^k + d^k)}{m^k(0) - m^k(d^k)} \\
&= \frac{-p^T d^k + \omega(\|c(x^k)\|_1 - \|c(x^k + d^k)\|_1)}{-p^T d^k + \omega(\|c(x^k)\|_1 - \|c(x^k) + \nabla c(x^k) d^k\|_1)} \\
&= 1 + \frac{-\omega \|c(x^k + d^k)\|_1}{-p^T d^k + \omega \|c(x^k)\|_1} \qquad \text{since } c(x^k) + \nabla c(x^k) d^k = 0, \\
&= 1 + \frac{-\omega \left\|c(x^k) + \nabla c(x^k) d^k + O(\|d^k\|^2)\right\|_1}{-p^T d^k + \omega \|\nabla c(x^k) d^k\|_1} \\
&= 1 + \frac{O(\|d^k\|^2)}{-p^T d^k + \omega \|\nabla c(x^k) d^k\|_1}.
\end{aligned}$$

We show now that the remainder term in this expression is of size $O(\|d^k\|)$ by showing that the denominator is bounded below by a multiple of $\|d^k\|$. From the optimality condition (22a), we have

$$p^T d^k + \omega(\lambda^*)^T \nabla c(x^*) d^k = (\mu^*)^T d^k + (\nu^*)^T d^k. \tag{31}$$

Since strict complementarity holds at $(x^*, \lambda^*, \mu^*, \nu^*)$, we have from the constant $\gamma > 0$ defined in (23) that

$$(\lambda^*)^T \nabla c(x^*) d^k \geq -(1-\gamma)\|\nabla c(x^*) d^k\|_1$$

and so

$$\left\|\nabla c(x^*) d^k\right\|_1 \geq \gamma \left\|\nabla c(x^*) d^k\right\|_1 - (\lambda^*)^T \nabla c(x^*) d^k. \tag{32}$$

Using (23) again, noting that $\mu_i^* > \gamma$ and $d_i^k \leq 0$ for $i \in \underline{\mathcal{A}}^*$, and $\nu_i^* < -\gamma$ and $d_i^k \geq 0$ for $i \in \overline{\mathcal{A}}^*$, we also have

$$-(\mu^*_{\underline{\mathcal{A}}^*})^T d^k_{\underline{\mathcal{A}}^*} - (\nu^*_{\overline{\mathcal{A}}^*})^T d^k_{\overline{\mathcal{A}}^*} \geq \gamma \left\|d^k_{\mathcal{A}^*}\right\|_1. \tag{33}$$

By combining (29), (31), (32), and (33), we have

$$\begin{aligned}
&-p^T d^k + \omega \left\|\nabla c(x^k) d^k\right\|_1 \\
&\quad = -p^T d^k + \omega \left\|\nabla c(x^*) d^k\right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad \geq -p^T d^k - \omega(\lambda^*)^T \nabla c(x^*) d^k + \omega\gamma \left\|\nabla c(x^*) d^k\right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad = -(\mu^*)^T d^k - (\nu^*)^T d^k + \omega\gamma \left\|\nabla c(x^*) d^k\right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad = -(\mu^*_{\underline{\mathcal{A}}^*})^T d^k_{\underline{\mathcal{A}}^*} - (\nu^*_{\overline{\mathcal{A}}^*})^T d^k_{\overline{\mathcal{A}}^*} + \omega\gamma \left\|\nabla c(x^*) d^k\right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad \geq \gamma \left\|d^k_{\mathcal{A}^*}\right\|_1 + \omega\gamma \left\|\nabla c(x^*) d^k\right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad = \gamma \left\| \begin{bmatrix} \omega \nabla c(x^*) \\ I_{\mathcal{A}^*} \end{bmatrix} d^k \right\|_1 + O(\left\|d^k\right\|^2) \\
&\quad \geq \gamma\zeta \left\|d^k\right\| + O(\left\|d^k\right\|^2)
\end{aligned}$$

where $\zeta > 0$ is a value related to the smallest singular value of $\begin{bmatrix} \omega \nabla c(x^*) \\ I_{\mathcal{A}^*} \end{bmatrix}$. By substituting this lower bound into the expression for $\rho^k(d^k)$ derived above, and using (29), we obtain

$$\rho^k = 1 + \frac{O(\left\|d^k\right\|^2)}{-p^T d^k + \omega \left\|\nabla c(x^k) d^k\right\|_1} = 1 + O(\left\|d^k\right\|) = 1 + O(\left\|x^* - x^k\right\|).$$

We can now argue recursively to obtain the result, in a manner whose first step is sketched informally here. By choosing x^0 sufficiently close to x^*, and Δ^0 sufficiently large, we can ensure that (a) the step d^0 defined as in Lemma 4 is does not reach the trust region bounds; (b) the ratio $\rho^0(d^0)$ is close enough to 1 to ensure that the step is accepted and that the trust-region radius is not decreased, that is, $\Delta^1 \geq \Delta^0$; (c) the step $x^1 = x^0 + d^0$ is sufficiently close to the solution that the conditions of Lemma 4 continue to be satisfied at the next iteration ($k = 1$) and that $\rho^1(d^1)$ continues to be close to 1.

Quadratic convergence follows from (29), since we have

$$\|x^{k+1} - x^*\| = \|x^k + d^k - x^*\| = O(\|x^k - x^*\|^2).$$

□

4 Accelerating Sℓ_1LP via an Active-Set Heuristic

We now consider the behavior of Sℓ_1LP as it approaches "underdetermined" solutions (see Definition 1). Since we can expect only linear convergence in these circumstances, we examine ways to accelerate the method by making use of active-set estimates and second-order information. In this section, we examine the elements of this approach in turn. We start with estimation of the active set in Subsection 4.1 and of values for the Lagrange multipliers in Subsection 4.2. Subsections 4.3 describes the nonlinear system of equations to be solved in the active-set strategy, while Subsection 4.4 describes a heuristic for modifying the active-set estimate if it appears to be faulty, and concludes with a full specification of the active-set heuristic. Subsection 4.5 describes how the active-set heuristic is inserted into the Sℓ_1LP algorithm, and discusses convergence properties of the enhanced approach.

Our discussion in this section refers both to the original formulation (1) and the nonsmooth penalty-function form (6). As we have noted, the two forms are equivalent for sufficiently large choice of penalty parameter ω, and it makes sense for a locally-convergent phase of the Sℓ_1LP algorithm to assume that the iterates have been steered toward a manifold where $c(x) = 0$ holds, so that the linearization of this condition can be enforced directly (rather than penalized) in computing the steps.

4.1 Active Set Identification

A vital ingredient of an active-set strategy for (1) include a reliable means for estimating those bounds that are active at the solution x^*. This issue has been examined in different contexts; see, for example Wright (1993); Hare and Lewis (2004); Lewis (2003). It has been shown that first-order information is often sufficient to make a reliable identification of the optimal active set, in certain circumstances. Oberlin and Wright (2006) have shown that the optimal active set of the nonsmooth-penalty formulation of nonlinear programming problem can be (approximately) identified by linear subproblems under certain conditions. They show if the current iterate x^k is close enough to x^* and the trust-region Δ^k is small enough, then the active set identified by $x^k + d^k$ is a subset of the optimal active set. Moreover, if LICQ and strict complementarity conditions hold, and if the trust-region radius Δ_k is large enough that $x^* - x^k$ is feasible for the subproblem yet small enough to prevent constraints inactive at x^* from becoming active in the subproblem, then the active set identified from the LP subproblem coincides with the optimal active set.

With this theory in mind, while not attempting to implement it rigorously, we use the simple heuristic of estimating the active set from the active set of the subproblem (15). Specifically, we estimate $\underline{\mathcal{A}}^*$ and $\overline{\mathcal{A}}^*$ by the sets $\underline{\mathcal{A}}_{k+1}$

and $\overline{\mathcal{A}}_{k+1}$, respectively, where $x^{k+1} = x^k + d^k$ and

$$\underline{\mathcal{A}}^{k+1} := \{i = 1, 2, \ldots, n \,|\, (x^k + d^k)_i = \underline{x}_i\},$$
$$\overline{\mathcal{A}}^{k+1} := \{i = 1, 2, \ldots, n \,|\, (x^k + d^k)_i = \overline{x}_i\},$$

as in (2). We observe that on Sℓ_1LP iterations, there are fewer and fewer changes to the active set as the iterations progress. It is therefore reasonable to use these sets as an estimate of of the optimal active sets once the number of changes drops below a specified threshold. Although these estimates may not be exact, they provide a good starting point for the heuristic discussed in Subsection 4.4, where incremental changes are tried for the active set in an attempt to restore consistency of the first-order optimality conditions.

4.2 Lagrange Multiplier Estimation

We now discuss how to estimate Lagrange multipliers λ, μ, and ν to initialize the active-set heuristic after an estimate of the active set becomes available. Because, as we discuss in Subsection 3.3, fast local convergence can be obtained without such estimates in the fully determined case, we consider here only *underdetermined* cases, for which $m + |\mathcal{A}^*| < n$. We assume that LICQ holds at x^*, that is, the constraint Jacobian matrix $\begin{bmatrix} \nabla c(x^*) \\ I_{\mathcal{A}^*} \end{bmatrix}$ has full row rank. The constraint Jacobian (that is, the basis matrix) for a nondegenerate solution of the LP subproblem (15) will contain n rows, with the extra $n - (m + |\mathcal{A}^*|)$ rows coming from enforcement of additional bound or trust-region constraints on d. We thus make the following assumption, to ensure that this augmented Jacobian retains its full-rank property.

Assumption 7 *All matrices of the form*

$$\begin{bmatrix} \nabla c(x^*) \\ I_{\mathcal{A}^*} \\ E^T \end{bmatrix}$$

are nonsingular where E is an $n \times (n - m - |\mathcal{A}^*|)$ matrix whose columns are drawn from $\{e_i \,|\, i \notin \mathcal{A}^*\}$. Thus there exists $\xi > 0$ such that for all such E we have

$$\left\| \begin{bmatrix} \nabla c(x^*) \\ I_{\mathcal{A}^*} \\ E^T \end{bmatrix}^{-1} \right\| \leq \xi.$$

Note that this assumption holds trivially for fully determined cases, since E is null in such cases.

The dual linear program for (15) is as follows:

$$\max_{\lambda,\mu,\nu} \quad \omega c(x^k)\lambda + (l^k)^T\mu + (u^k)^T\nu \tag{34a}$$

$$\text{subject to} \quad p + \omega\nabla c(x^k)^T\lambda - \mu - \nu = 0 \tag{34b}$$

$$-\mathbf{1} \le \lambda \le \mathbf{1}, \quad \mu \ge 0, \quad \nu \le 0, \tag{34c}$$

where

$$l^k := \max(\underline{x} - x^k, -\Delta^k), \qquad u^k := \min(\overline{x} - x^k, -\Delta^k).$$

The following theorem states the relation between the solution of (34) and the optimal Lagrange multipliers $(\lambda^*, \mu^*, \nu^*)$ under Assumption 7. This result rests on an assumption that identification of the active set via the penalized linear programming model, as described in Oberlin and Wright (2006), has given an accurate result.

Theorem 8 *Suppose that $(x^*, \lambda^*, \mu^*, \nu^*)$ satisfies (22) and the optimal active set of (1) is identified by $x^k + d^k$. That is, we have*

$$(x^k + d^k)_i = \underline{x}_i \quad \Leftrightarrow \quad i \in \underline{\mathcal{A}}^*, \qquad (x^k + d^k)_i = \overline{x}_i \quad \Leftrightarrow \quad i \in \overline{\mathcal{A}}^*.$$

If the LP subproblem (15) has a solution $(d, \alpha) = (d^k, 0)$ which is not degenerate and the Assumption 7 holds, then provided that x^k is sufficiently close to x^*, the dual problem (34) has a solution $(\lambda^k, \mu^k, \nu^k)$ such that

$$(\lambda^k, \mu^k, \nu^k) = (\lambda^*, \mu^*, \nu^*) + O(\|x^* - x^k\|).$$

Proof Let $\underline{\mathcal{A}}_\Delta^{k+1}$ and $\overline{\mathcal{A}}_\Delta^{k+1}$ be defined by

$$\underline{\mathcal{A}}_\Delta^{k+1} = \left\{ i = 1, 2, \ldots, n \,\middle|\, (x^k + d^k)_i = \underline{x}_i \right\} \cup \left\{ i = 1, 2, \ldots, n \,\middle|\, d_i^k = -\Delta^k \right\},$$
$$\overline{\mathcal{A}}_\Delta^{k+1} = \left\{ i = 1, 2, \ldots, n \,\middle|\, (x^k + d^k)_i = \overline{x}_i \right\} \cup \left\{ i = 1, 2, \ldots, n \,\middle|\, d_i^k = \Delta^k \right\}.$$

By the assumptions of the theorem, we have $\underline{\mathcal{A}}^* \subset \underline{\mathcal{A}}_\Delta^{k+1}$ and $\overline{\mathcal{A}}^* \subset \overline{\mathcal{A}}_\Delta^{k+1}$. From linear programming duality, we have

$$\mu_i^k = 0 \ \text{ for } \ i \notin \underline{\mathcal{A}}_\Delta^{k+1}, \qquad \nu_i^k = 0 \ \text{ for } \ i \notin \overline{\mathcal{A}}_\Delta^{k+1}.$$

We know further that

$$\mu_i^* = 0 \ \text{ for } \ i \in \underline{\mathcal{A}}_\Delta^{k+1} \backslash \underline{\mathcal{A}}^*, \qquad \nu_i^* = 0 \ \text{ for } \ i \in \overline{\mathcal{A}}_\Delta^{k+1} \backslash \overline{\mathcal{A}}^*.$$

Then from (22a) and (34b), we have

$$\begin{aligned}
\begin{bmatrix}\nabla c(x^k)^T & I^T_{\underline{\mathcal{A}}^{k+1}_\Delta} & I^T_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix}
\begin{bmatrix}\omega\lambda^k \\ -\mu^k_{\underline{\mathcal{A}}^{k+1}_\Delta} \\ -\nu^k_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix} &= -p \\
&= \begin{bmatrix}\nabla c(x^*)^T & I^T_{\underline{\mathcal{A}}^*} & I^T_{\overline{\mathcal{A}}^*}\end{bmatrix}
\begin{bmatrix}\omega\lambda^* \\ -\mu^*_{\underline{\mathcal{A}}^*} \\ -\nu^*_{\overline{\mathcal{A}}^*}\end{bmatrix} \\
&= \begin{bmatrix}\nabla c(x^*)^T & I^T_{\underline{\mathcal{A}}^{k+1}_\Delta} & I^T_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix}
\begin{bmatrix}\omega\lambda^* \\ -\mu^*_{\underline{\mathcal{A}}^{k+1}_\Delta} \\ -\nu^*_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix} \\
&= \begin{bmatrix}\nabla c(x^k)^T & I^T_{\underline{\mathcal{A}}^{k+1}_\Delta} & I^T_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix}
\begin{bmatrix}\omega\lambda^* \\ -\mu^*_{\underline{\mathcal{A}}^{k+1}_\Delta} \\ -\nu^*_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix} + O(\|x^k - x^*\|).
\end{aligned}$$

Since the solution $(d^k, 0)$ is not degenerate, and by smoothness of c, we have from Assumption 7 that the matrix $\begin{bmatrix}\nabla c(x^k) \\ I_{\underline{\mathcal{A}}^{k+1}_\Delta \cup \overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix}^T$ is a square, uniformly nonsingular matrix for x^k sufficiently close to x^*, and thus

$$\begin{bmatrix}\omega\lambda^k \\ -\mu^k_{\underline{\mathcal{A}}^{k+1}_\Delta} \\ -\nu^k_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix} = \begin{bmatrix}\omega\lambda^* \\ -\mu^*_{\underline{\mathcal{A}}^{k+1}_\Delta} \\ -\nu^*_{\overline{\mathcal{A}}^{k+1}_\Delta}\end{bmatrix} + O(\|x^k - x^*\|).$$

Since $\mu^*_i = \mu^k_i = 0$ for $i \notin \underline{\mathcal{A}}^{k+1}_\Delta$, and $\nu^*_i = \nu^k_i = 0$ for $i \notin \overline{\mathcal{A}}^{k+1}_\Delta$, we have

$$\begin{bmatrix}\lambda^k \\ \mu^k \\ \nu^k\end{bmatrix} = \begin{bmatrix}\lambda^* \\ \mu^* \\ \nu^*\end{bmatrix} + O(\|x^* - x^k\|),$$

as desired. □

Theorem 8 indicates that the dual solution of the LP subproblem can be used as a good approximation of the optimal Lagrange multipliers of the original CNSO problem (6), for x^k close to x^*. However, the estimated Lagrange multipliers may violate complementarity conditions in the optimality conditions (22), due to the presence of trust-region constraints. To fix these violations, we drop the values of dual solution corresponding to the trust-region constraints and define $(\lambda^{k+1}, \mu^{k+1}, \nu^{k+1})$ to start the active-set heuristic at iterate $x^{k+1} = x^k + d^k$ as below:

$$\lambda^{k+1} = \lambda^k, \quad \mu^{k+1}_i = \begin{cases}\mu^k_i & i \in \underline{\mathcal{A}}^{k+1}, \\ 0 & \text{otherwise,}\end{cases} \quad \nu^{k+1}_i = \begin{cases}\nu^k_i & i \in \overline{\mathcal{A}}^{k+1}, \\ 0 & \text{otherwise}.\end{cases} \tag{35}$$

4.3 The Active-Set Heuristic

The active-set heuristic consists of a sequence of Newton step on a system of nonlinear equations like (7), with $\underline{\mathcal{A}}^*$ and $\overline{\mathcal{A}}^*$ replaced by current estimates of these active sets, and the Newton linearization taking place around the latest primal-dual iterate $(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})$, which is initialized to the primal-dual point $(x^k + d^k, \lambda^{k+1}, \mu^{k+1}, \nu^{k+1})$ obtained as in the previous subsection. We initialize the active set estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$ to $\underline{\mathcal{A}}^{k+1}$ and $\overline{\mathcal{A}}^{k+1}$, respectively, and define

$$\underline{\mathcal{I}}' = \{1, 2, \ldots, n\} \backslash \underline{\mathcal{A}}', \quad \overline{\mathcal{I}}' = \{1, 2, \ldots, n\} \backslash \overline{\mathcal{A}}', \quad \mathcal{I}' = \{1, 2, \ldots, n\} \backslash (\underline{\mathcal{A}}' \cup \overline{\mathcal{A}}').$$

as in (3). The Newton equations are as follows:

$$\begin{bmatrix} \tilde{H} & \tilde{A} & -I & -I \\ \tilde{A}^T & & & \\ I_{\underline{\mathcal{A}}',\cdot} & & & \\ I_{\overline{\mathcal{A}}',\cdot} & & & \\ & I_{\underline{\mathcal{I}}',\cdot} & & \\ & & & I_{\overline{\mathcal{I}}',\cdot} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\mu \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) \\ c(\tilde{x}) \\ (\tilde{x} - \underline{x})_{\underline{\mathcal{A}}'} \\ (\tilde{x} - \overline{x})_{\overline{\mathcal{A}}'} \\ \tilde{\mu}_{\underline{\mathcal{I}}'} \\ \tilde{\nu}_{\overline{\mathcal{I}}'} \end{bmatrix} \tag{36}$$

where

$$\tilde{H} = \sum_{i=1}^{m} \tilde{\lambda}_i \nabla^2 c_i(\tilde{x}), \quad \tilde{A} = \nabla c(\tilde{x})^T.$$

(Note that the coefficient matrix in (36) is square.) Denoting the solution of (36) by $(\Delta x', \Delta\lambda', \Delta\mu', \Delta\nu')$, we define a provisional estimate of the next iterate by

$$(x', \lambda', \mu', \nu') = (\tilde{x}, \tilde{\lambda}, \tilde{\nu}, \tilde{\mu}) + (\Delta x', \Delta\lambda', \Delta\mu', \Delta\nu'). \tag{37}$$

We accept $(x', \lambda', \mu', \nu')$ as the new iterate only if the following set of inequalities (analogous to (8)) hold:

$$\underline{x}_i < x'_i < \overline{x}_i \;\text{ for }\; i \in \mathcal{I}', \tag{38a}$$

$$\mu'_i \geq 0 \;\text{ for }\; i \in \underline{\mathcal{A}}', \tag{38b}$$

$$\nu'_i \leq 0 \;\text{ for }\; i \in \overline{\mathcal{A}}'. \tag{38c}$$

We require in addition that the new iterate improves the norm of the algebraic KKT conditions (4a), (4b) by at least a factor of 1/2, that is,

$$\left\| \begin{bmatrix} \nabla_x \mathcal{L}(x', \lambda', \mu', \nu') \\ c(x') \end{bmatrix} \right\| \leq \frac{1}{2} \left\| \begin{bmatrix} \nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}, \tilde{\mu}) \\ c(\tilde{x}) \end{bmatrix} \right\|. \tag{39}$$

If both these conditions hold, we replace $(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})$ by $(x', \lambda', \mu', \nu')$, leave the active-set estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$ unchanged, and take another active-set step.

If condition (38) fails to hold, we try to "tweak" the active set and resolve the Newton equations for the new active sets, as described in the next subsection.

Note that the system (36) contains a great deal of structure. By performing various block eliminations and substitutions, we can reduce it significantly in size. Some of the variables can be obtained directly, as follows

$$\begin{aligned} \Delta x'_{\underline{\mathcal{A}}'} &= (\underline{x} - \tilde{x})_{\underline{\mathcal{A}}'}, \\ \Delta x'_{\overline{\mathcal{A}}'} &= (\overline{x} - \tilde{x})_{\overline{\mathcal{A}}'}, \\ \Delta \mu'_{\underline{\mathcal{I}}'} &= -\mu_{\underline{\mathcal{I}}'}, \\ \Delta \nu'_{\overline{\mathcal{I}}'} &= -\nu_{\overline{\mathcal{I}}'}. \end{aligned} \tag{40}$$

There are more variables that appear in just one equation, whose values can be defined as follows:

$$\begin{aligned} \Delta \mu'_{\underline{\mathcal{A}}'} &= (\tilde{H}\Delta x' + \tilde{A}\Delta\lambda' + \nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}, \tilde{\mu}))_{\underline{\mathcal{A}}'}, \\ \Delta \nu'_{\overline{\mathcal{A}}'} &= (\tilde{H}\Delta x' + \tilde{A}\Delta\lambda' + \nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}, \tilde{\mu}))_{\overline{\mathcal{A}}'}. \end{aligned} \tag{41}$$

The remaining variables can be obtained by solving the following linear system:

$$\begin{bmatrix} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & \end{bmatrix} \begin{bmatrix} \Delta x'_{\mathcal{I}'} \\ \Delta\lambda' \end{bmatrix} = - \begin{bmatrix} [\nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}, \tilde{\mu})]_{\mathcal{I}'} \\ c(\tilde{x}) \end{bmatrix}. \tag{42}$$

The coefficient matrix in (42) is symmetric indefinite, so we can perform an factorization involving a lower triangular matrix L and a block diagonal matrix D:

$$LDL^T = \begin{bmatrix} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & \end{bmatrix}. \tag{43}$$

Having calculated this factorization, the solution of (42) can be computed by performing two triangular substitutions and some other simple operations.

4.4 Active Set Adjustment

In Subsection 4.1, we discussed taking the active set from the latest LP subproblem as our estimate of the optimal active set. We may find that the step produced in (36), (37) does not satisfy the inequality conditions (38), which may be an indication that the current active set estimate is not optimal. Since it is probably not too different from the optimum, however, we propose another heuristic for making a few changes to it, rather than discarding the step completely and returning to the Sℓ_1LP algorithm.

The following rules are used to modify the active-set estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$.

- If $x'_i < \underline{x}_i$, we add i to $\underline{\mathcal{A}}'$; if $x'_i < \overline{x}_i$, we add i to $\overline{\mathcal{A}}'$.
- If $\mu'_i < 0$, we remove i from $\underline{\mathcal{A}}'$.
- If $\nu'_i > 0$, we remove i from $\underline{\mathcal{A}}'$.

Following these changes, we calculate the Newton step from (36), with the modified sets $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$, recalculate the provisional new iterate $(x', \lambda', \mu', \nu')$ from (37), and retest the inequality conditions (38). We declare "success" when the inequality conditions are satisfied and the sufficient decrease test (39) holds. Otherwise, we repeat the adjustment procedure at most ten times. We declare "failure" and return to Sℓ_1LP iterations if

- the conditions (38) are not satisfied at any of these ten steps,
- we re-encounter estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$ that had been tried already at one of the previous steps, or
- the total number of components in the active set $(|\underline{\mathcal{A}}'| + |\overline{\mathcal{A}}'|)$ grows larger than $n - m$.

The second condition is invoked to prevent cycling among a set of choices for the active-set estimates. When the third condition holds, the solution is "overdetermined" by its constraints, and the coefficient matrix in (36) becomes structurally singular.

Rather that solving (36) from scratch after modification of the active sets, we re-use the factorization (43), modifying the system (42) by adding columns to account for the changes to the active-set estimates. Any revised problem with the altered active set approximation can be written in the following form:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & \mathrm{V} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & \\ \hline X^T & & S \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline \cdot \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})]_{\mathcal{I}'} \\ (\tilde{x}) \\ \hline s \end{array}\right] \tag{44}$$

where the matrices X, V, S and the vector s capture the changes to the active set. Below, we describe the five possible ways in which the active set can be altered, and show how to define the system (44) to account for these cases.

(a) When x_i, for some $i \notin \mathcal{A}'$, moves from the interior of its box constraint to beyond the lower bound. Since the constraint $x_i \geq \underline{x}_i$ needs to be active, the value of Δx_i should be $\underline{x}_i - x_i^k$. Also the corresponding Lagrange multiplier μ_i is allowed to move away from zero. The modified linear system can thus be defined as follows:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & (e_i)_{\mathcal{I}'} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & \\ \hline (e_i)^T_{\mathcal{I}'} & & \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline -\Delta\mu_i \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})]_{\mathcal{I}'} \\ c(\tilde{x}) \\ \hline \tilde{x}_i - \underline{x}_i \end{array}\right].$$

(b) When x_i, for some $i \notin \mathcal{A}'$, moves from the interior of its box constraint to the upper bound, a similar construction yields the following modified system:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & (e_i)_{\mathcal{I}'} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & \\ \hline (e_i)^T_{\mathcal{I}'} & & \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline -\Delta\nu_i \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})]_{\mathcal{I}'} \\ c(\tilde{x}) \\ \hline \tilde{x}_i - \overline{x}_i \end{array}\right].$$

(*c*) When, for some $i \in \mathcal{A}'$, a component ν_i or μ_i moves from one nonzero value to another nonzero value with a different sign, we take it as an indication that the corresponding component of x should move away from its bound. We thus set the Lagrange multiplier component in question to zero, and allow Δx_i to become nonzero, The augmented linear system has the following form:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & \tilde{H}_{\mathcal{I}',i} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & (\tilde{A}_{i,\cdot})^T \\ \hline \tilde{H}_{i,\mathcal{I}'} & \tilde{A}_{i,\cdot} & \tilde{H}_{i,i} \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline \Delta x_i \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x},\tilde{\lambda},\tilde{\mu},\tilde{\nu})]_{\mathcal{I}'} \\ c(\tilde{x}) \\ \hline [\nabla_x \mathcal{L}(\tilde{x},\tilde{\lambda},\tilde{\mu},\tilde{\nu})]_i \end{array}\right].$$

If we permute the rows and columns of this system, we can recover a structure similar to (36).

(*d*) When, for some $i \in \overline{\mathcal{A}}'$, x_i moves from its upper bound to its lower bound, we redefine $\Delta x_i = \underline{x}_i - \overline{x}_i$, and define the augmented system as follows:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & \tilde{H}_{\mathcal{I}',i} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & (\tilde{A}_{i,\cdot})^T \\ \hline & & 1 \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline \Delta x_i \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x},\tilde{\lambda},\tilde{\mu},\tilde{\nu})]_{\mathcal{I}'} \\ c(\tilde{x}) \\ \hline \overline{x}_i - \underline{x}_i \end{array}\right].$$

(*e*) When, for some $i \in \underline{\mathcal{A}}'$, x_i moves from its lower bound to its upper bound, we modify the augmented system as follows:

$$\left[\begin{array}{cc|c} \tilde{H}_{\mathcal{I}',\mathcal{I}'} & \tilde{A}_{\mathcal{I}',\cdot} & \tilde{H}_{\mathcal{I}',i} \\ (\tilde{A}_{\mathcal{I}',\cdot})^T & & (\tilde{A}_{i,\cdot})^T \\ \hline & & 1 \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathcal{I}'} \\ \Delta\lambda \\ \hline \Delta x_i \end{array}\right] = - \left[\begin{array}{c} [\nabla_x \mathcal{L}(\tilde{x},\tilde{\lambda},\tilde{\mu},\tilde{\nu})]_{\mathcal{I}'} \\ c(\tilde{x}) \\ \hline \underline{x}_i - \overline{x}_i \end{array}\right].$$

These modifications can be combined when multiple changes to the active set are made on a single step. Each such change results in one extra row and column being added to the reduced augmented system. Appendix A contains further details on how the factorization (43) for the original coefficient matrix can be leveraged to solve the systems above efficiently.

The full specification of the active-set heuristic appears as Algorithm 2.

4.5 Inserting the Active-Set Heuristic into the Sℓ_1LP Algorithm

The active-set heuristic can be invoked after an iteration of the Sℓ_1LP algorithm is completed, if we decide that the active sets $\underline{\mathcal{A}}^k$ and $\overline{\mathcal{A}}^k$ have settled down (for example, they have changed by fewer than ten components). To invoke the active-set heuristic after a successful iteration k of Sℓ_1LP, we estimate Lagrange multipliers according to the procedure of Subsection 4.2, then invoke Algorithm 2 with $(\tilde{x},\tilde{\lambda},\tilde{\mu},\tilde{\nu}) = (x^k + d^k, \lambda^{k+1}, \mu^{k+1}, \nu^{k+1})$, $\underline{\mathcal{A}}' = \underline{\mathcal{A}}^{k+1}$, and $\overline{\mathcal{A}}' = \overline{\mathcal{A}}^{k+1}$.

If Algorithm 2 exits with a failure of the active-set heuristic, we simply return to the Sℓ_1LP algorithm and pick up where we left off, with the latest

Algorithm 2 Active-Set Heuristic

Require:
Primal-dual iterate $(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})$ and active set estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$;
Ensure:
Either declare failure and return to iterating in Algorithm 1, or produce a primal-dual solution estimate $(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu})$;

loop
  counter$\leftarrow 0$
  loop
    counter$\leftarrow$counter$+1$;
    if counter > 10 **then**
      stop and return to Algorithm 1; ▶ No suitable active set found
    end if
    Calculate candidate iterate $(x', \lambda', \mu', \nu')$ from (36) and (37);
    if Conditions (38) are satisfied **then**
      if (39) holds **then**
        break; ▶ Successful active-set iteration
      else
        stop and return to Algorithm 1; ▶ Insufficient improvement
      end if
    else
      Apply heuristic adjustment process of Subsection 4.4 to obtain modified active set estimates $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$;
      if The adjusted sets $\underline{\mathcal{A}}'$ and $\overline{\mathcal{A}}'$ were encountered on a previous iteration of this loop OR $|\underline{\mathcal{A}}'| + |\overline{\mathcal{A}}'| > n - m$ **then**
        stop and return to Algorithm 1; ▶ No suitable active set found.
      end if
    end if
  end loop
  Set $(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) \leftarrow (x', \lambda', \mu', \nu')$;
  if $\left\| \begin{bmatrix} \nabla_x \mathcal{L}(\tilde{x}, \tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) \\ c(\tilde{x}) \end{bmatrix} \right\| \leq$ **tol then**
    stop; ▶ Success!
  end if
end loop

values of x^k and Δ^k, and proceed with further iterations according to Algorithm 1. Thus, it is clear that the global convergence theory developed in Section 3 continues to hold.

Otherwise, if we remain in the active-set heuristic indefinitely, the conditions (38) together with the decrease condition (39) ensure that any limit point of this heuristic satisfies the first-order optimality conditions (4) of (1). Although we refrain from offering a formal theory for the rate of convergence, it is clear that if the active-set heuristic is invoked in the neighborhood of a point x^* at which first-order conditions, LICQ, strict complementarity, and second-order sufficient conditions for (1) are satisfied, it can be expected to converge quadratically, provided that c is twice continuously differentiable near x^*.

5 Feasibility Restoration for Power Systems

In this section we outline the problem of restoring feasibility to a power network by optimal load shedding. The AC power flow model is defined in Subsection 5.1, while we formulate the optimal load-shedding problem as a problem of the form (1) in Subsection 5.2.

We make use throughout this section of the following notations:

- Set of buses: $\mathcal{N}$
- Set of generators: $\mathcal{G} \subseteq \mathcal{N}$
- Set of demand buses: $\mathcal{D} \subseteq \mathcal{N}$
- Index of the slack (reference) bus: $s \in \mathcal{N}$
- Set of lines: $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$
- Unit imaginary number: j
- Complex power at bus $i \in \mathcal{N}$: $P_i + jQ_i$. (P_i is the active power and Q_i the reactive power).
- Complex voltage at bus $i \in \mathcal{N}$: $V_i e^{j\theta_i}$. (V_i is the voltage magnitude and θ_i is the voltage angle.)
- Difference of θ_i and θ_j: $\theta_{ij} = \theta_i - \theta_j$
- Admittance for line (i,k) (that is, the (i,k) element of the line admittance matrix): $G_{ik} + jB_{ik}$.

We note that $\{\mathcal{G}, \mathcal{D}, \{s\}\}$ is a partition of $\mathcal{N}$, i.e. the sets $\mathcal{G}, \mathcal{D}$ and $\{s\}$ are mutually disjoint and $\mathcal{D} \cup \mathcal{G} \cup \{s\} = \mathcal{N}$.

5.1 Power Flow Problems

To operate a power system we need to know the voltages and powers at all buses, captured in the vector (V, θ, P, Q), that satisfy the following power-flow balance equations arising from Kirchhoff's laws:

$$\begin{bmatrix} F^P(V,\theta) \\ F^Q(V,\theta) \end{bmatrix} = 0, \tag{45}$$

where the ith entries of F^P and F^Q are nonlinear functions defined as

$$F_i^P(V,\theta) := V_i \sum_{k:(i,k)\in\mathcal{L}} V_k (G_{ik}\cos(\theta_{ik}) + B_{ik}\sin(\theta_{ik})) - P_i \tag{46a}$$

$$F_i^Q(V,\theta) := V_i \sum_{k:(i,k)\in\mathcal{L}} V_k (G_{ik}\sin(\theta_{ik}) - B_{ik}\cos(\theta_{ik})) - Q_i. \tag{46b}$$

The power flow problem is to find a solution (V, θ, P, Q) to the above equations, where two out of each quadruplet $(V_i, \theta_i, P_i, Q_i)$ are known for each bus. The known quantities are varied by the type of bus. For generators $i \in \mathcal{G}$, voltage magnitude V_i and the active power P_i that the generators can produce are known. For the slack bus s, the voltage magnitude V_s and angle θ_s are known.

For load buses $i \in \mathcal{D}$, the active power P_i, and reactive power Q_i are known. Since Q_i for $i \in \mathcal{G}$ can be determined directly from (46b), we can reduce (45) to a system of $2|\mathcal{D}| + |\mathcal{G}|$ nonlinear equations with $2|\mathcal{D}| + |\mathcal{G}|$ variables $(V_{\mathcal{D}}, \theta_{\mathcal{G}\cup\mathcal{D}})$:

$$F(V,\theta) = \begin{bmatrix} F_{\mathcal{G}}^{P}(V,\theta) \\ F_{\mathcal{D}}^{P}(V,\theta) \\ F_{\mathcal{D}}^{Q}(V,\theta) \end{bmatrix} = 0, \tag{47}$$

where V_s, θ_s, $V_{\mathcal{G}}$, $P_{\mathcal{G}}$, $P_{\mathcal{D}}$ and $Q_{\mathcal{D}}$ are given.

Newton's method is widely used for solving (47). If a problem is well-conditioned and a sufficiently good initial starting point is given, it converges to a solution in a few iterations. Moreover, since first derivatives can be computed easily for this system, and the Jacobian is quite sparse, it can be implemented efficiently.

More robust methods to solve ill-conditioned or badly-initialized power flow problems have been studied by several authors (Iwamoto and Tamura 1981; Tripathy et al 1982; Milano 2009), including damped Newton methods and homotopy methods. There is no known Newton-based method that can detect reliably the unsolvability of a power flow problem, but in practice, failure of Newton's methods from a reasonable starting point is strong evidence of nonexistence of a solution. A method based on Semidefinite programming (SDP), which is studied by Lavaei and Low (2012) for OPF problems, has been proposed recently as a more rigorous means of finding solutions and proving nonexistence, but it cannot resolve all cases (see Molzahn et al (2013)).

5.2 Formulation of Feasibility Restoration for Power Systems

Since we are interested in cases in which a solution of (47) may not exist, we first consider reformulating these nonlinear equations as an optimization problem, as follows:

$$\min_{V_{\mathcal{D}},\theta_{\mathcal{G}\cup\mathcal{D}}} \quad \|F(V,\theta)\|_1 ,$$

which can be reformulated in the framework (1) with the use of auxiliary variables q and r, as follow:

$$\begin{aligned} \min_{V_{\mathcal{D}},\theta_{\mathcal{D}\cup\mathcal{G}},q,r} \quad & e^T(q+r) \\ \text{subject to} \quad & F(V,\theta) - (q-r) = 0, \quad q,r \geq 0, \end{aligned} \tag{48}$$

where e is a vector of all 1's. This formulation has practical interest, as it indicates the amount by which active powers P_i, $i \in \mathcal{D} \cup \mathcal{G}$ and reactive powers Q_i, $i \in \mathcal{D}$ need to be adjusted in order to restore feasibility to the network. The use of an ℓ_1-norm objective promotes sparsity of the nonzero entries, which usually means that a relatively small number of loads need to be adjusted to recover feasibility. The ℓ_1-norm formulation also lends itself well to the linear-programming-based techniques described in this paper.

A number of previous works (Barboza and Salgado 2001b; Overbye 1995) measure the distance to the operational region with the ℓ_2-norm (Euclidean distance), but this does not lead to a satisfiactory operational solution, since it generally requires adjusting loads at many demand nodes.

While the formulation (48) captures the basic concept of feasibility restoration, we need to refine it by taking certain practicalities into account. Power flow problem formulations often omit restrictions on voltage magnitudes on buses, since these do not usually enter into consideration under normal conditions. Since we are dealing with stressed and disrupted networks here, it makes good sense to include them in out formulations, to exclude solutions that would not be operational by restricting the voltages to a certain range. Another factor to consider, since we intend the formulation to provide a practical indication of how feasibility can be restored, is to impose practical constraints on the adjustments indicated by the model. For example, when load-shedding is needed on a demand bus, the amount of the load-shedding should not exceed the total amount of demand at the bus. Also, since the active and reactive powers on a bus are closely related, they should be adjusted by the same fraction. With these additional restrictions, we obtain the following enhanced form of (48):

$$\min_{\substack{V_{\mathcal{D}},\theta_{\mathcal{D}\cup\mathcal{G}},\\ \sigma^+_{\mathcal{G}},\sigma^-_{\mathcal{G}},\rho_{\mathcal{D}}}} \quad \sum_{i\in\mathcal{G}} |P_i|\,(\sigma_i^+ + \sigma_i^-) + \sum_{i\in\mathcal{D}} |P_i|\,\rho_i \tag{49a}$$

$$\text{subject to} \quad F_i^P(V,\theta) - |P_i|\,(\sigma_i^+ - \sigma_i^-) = 0 \quad i\in\mathcal{G} \tag{49b}$$

$$F_i^P(V,\theta) - |P_i|\,\rho_i = 0 \quad i\in\mathcal{D} \tag{49c}$$

$$F_i^Q(V,\theta) - |Q_i|\,\rho_i = 0 \quad i\in\mathcal{D} \tag{49d}$$

$$\underline{V} \le V_i \le \overline{V} \quad i\in\mathcal{D} \tag{49e}$$

$$0 \le \sigma_i^+ \le \overline{\sigma}_i^+ \quad i\in\mathcal{G} \tag{49f}$$

$$0 \le \sigma_i^- \le \overline{\sigma}_i^- \quad i\in\mathcal{G} \tag{49g}$$

$$0 \le \rho_i \le \overline{\rho}_i \quad i\in\mathcal{D}, \tag{49h}$$

where $\underline{V}$ and $\overline{V}$ are the lower and upper limits on voltage magnitudes of demand buses, $\overline{\sigma}_i^{\pm}$, $i\in\mathcal{G}$ are the bounds on the active power adjustments of generators and $\overline{\rho}_i$, $i\in\mathcal{D}$ are bounds on the active and reactive power adjustments of demand nodes. We have assumed here that $P_i \le 0$ and $Q_i \le 0$ for $i\in\mathcal{D}$, and $P_i > 0$ for $i\in\mathcal{G}$, following convention. We note the following points.

- The problem has the form (1) for which we describe and analyze algorithms in earlier sections.
- Constraints (49b), (49c), and (49d) represent relaxations of the power flow equations (47) in which the loads $P_{\mathcal{G}}$, $P_{\mathcal{D}}$, and $Q_{\mathcal{D}}$ are modified by a certain relative amount, captured by the variables σ_i^+, σ_i^-, and ρ_i.
- Constraint (49b) ensures that power generation can be either increased or decreased, but (49c) and (49d) ensure that loads at demand nodes can only decrease.

– The same variable ρ_i is used in the active and reactive power balance equations (49c) and (49d), since active and reactive load shedding should occur in the same fraction.
– Box constraints on the load shedding variables (49f), (49g), and (49h) ensure that adjustments cannot exceed user-defined limits. (Upper bounds $\overline{\sigma}_i^+$, $\overline{\sigma}_i^-$, and $\overline{\rho}_i$ should not exceed 1.)
– The bounds (49e) guarantee that voltage levels are operationally viable in the solution.
– In the objective (49a), only active power adjustments are considered.

The feasibility restoration problem (49) is neither linear nor convex, so we can guarantee only a local solution. The problem generalizes (47) in that if a solution of the latter problem exists, it will yield a global solution of (49) with an objective of zero when we set $\sigma_i^+ = \sigma_i^- = 0$ for $i \in \mathcal{G}$ and $\rho_i = 0$ for $i \in \mathcal{D}$, provided the voltage constraints (49e) are satisfied. Moreover, by the well-known sparsity property induced by the ℓ_1 objective, we expect few of the components of $\sigma_{\mathcal{G}}^+$, $\sigma_{\mathcal{G}}^-$, and $\rho_{\mathcal{D}}$ to be nonzero at a typical solution of (49).

The solution of (49) suggests to the grid operator a feasible operating point and a load-shedding pattern that can be used to attain this point.

6 Experimental Results

In this section we present the experimental results for our Sℓ_1LP algorithm applied to the formulation (49) of power system feasibility restoration. The algorithm is implemented in MATLAB[1] on MacBook Pro (2 GHz Intel Core i7 with 8GB RAM) with CPLEX[2] as a linear programming solver. Specifically, the CPLEX Class API for MATLAB is used to exploit the warm start feature of LP solver. The MATLAB functions provided by MATPOWER[3] (Zimmerman et al 2011) are used to read power systems data and to compute the values required to formulate the problems (such as the admittance matrix and the derivatives of the power flow equations). The data sets included in MATPOWER package, which contains the power systems data from Power Systems Test Case Archive (University of Washington, Electrical Engineering 2014), are used for the experiments, but they are modified to create infeasible instances. (The original data is feasible for the power flow problem, so leads to a zero feasible objective in (49).) Our implementation includes pre-compiled code (CPLEX) and MATLAB code. We compare it with IPOPT[4] (Wächter and Biegler 2006), an interior-point solver written in `C++` and compiled for MATLAB with the MA27 linear solver; and MIPS, a MATPOWER Interior Point Solver written in MATLAB code.

[1] Version 8.1.0.604 (R2013a)
[2] Version 12.6
[3] Version 4.1
[4] Version 3.11.7

Bus			Lines	Variables				Constraints		$(\underline{V}, \overline{V})$
Ref	$\mathcal{G}$	$\mathcal{D}$		V	θ	$\sigma^{\pm}, \rho$	Total	$=$	$\leq$ or $\geq$	
1	6	50	80	56	50	37	143	106	186	(0.93, 1.07)

(a) Information for the IEEE 57 Bus System

β	$n - m - \lvert\mathcal{A}^*\rvert$	Iterations		Time (sec)	Objective Value	Buses w/ Loads Shed
		Sℓ_1LP	AS			
1.0	0	3(0)	-	0.03	0.00e-00	0
1.2	0	4(2)	-	0.04	2.93e-02	2
1.4	0	4(5)	-	0.04	8.37e-02	4
1.6	0	16(34) 5(18)	- 3(0)	0.10 0.06	1.70e-01	8
1.8	0	5(14)	-	0.04	2.71e-01	9
2.0	4	34(119) 3(26)	- 3(2)	0.19 0.05	3.56e-01	11

(b) Results on IEEE 57 Bus System

Table 1: IEEE 57 Bus System

For the Sℓ_1LP phase, the following values are used for the parameters that appear in Algorithm 1 and (16):

$$\underline{\eta} = 0.5, \quad \overline{\eta} = 0.25, \quad \underline{\rho} = 0.1, \quad \epsilon = 10^{-3}, \quad c_1 = 2, \quad c_2 = 0.5.$$

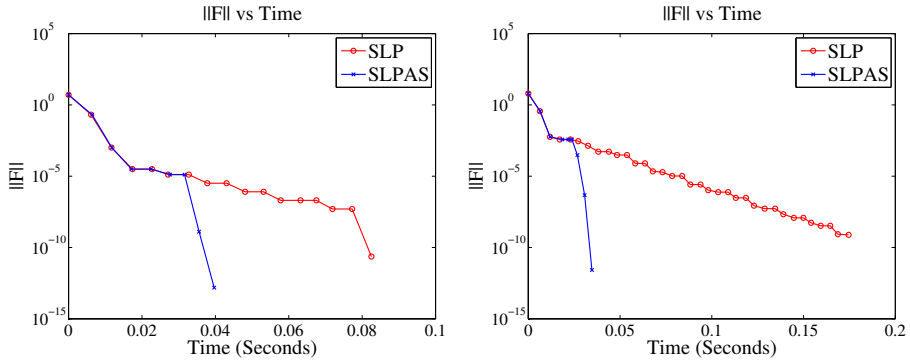
The tolerance to check the activities and violations of constraints is set to 10^{-9}. For each LP subproblem, the dual simplex option is used in CPLEX.

For IPOPT and MIPS, all parameters are set to their default values.

6.1 IEEE 57-Bus System (`case57.m`) with Line Impedance Perturbations

We study a standard test case, the IEEE 57-bus system, to explore the overall behavior of our algorithm. Information about the system is provided in Table 1a. We modify the specifications of the problem by multiplying the impedance of each line by factor β in the range $[1, 2]$ to disrupt the system for the experiments. The values given in MATPOWER data file are used as the initial starting point for the algorithm.

Table 1b shows the results for the each value of β. The column "$n - m - |\mathcal{A}^*|$" indicates whether the solution is fully determined by its constraints (the value is zero) or not (a positive integer). For the fully-determined cases, the active set heuristic is often not invoked, in which case we show the results as a single line. In cases for which the active-set heuristics is used, we show two lines of results, one for the case in which this heuristic is turned off and one for the case in which we allow it to be invoked. The columns labelled Sℓ_1LP and AS show the number of iterations, with the total number of dual simplex iterations in parentheses.



(a) $\beta = 1.6$ (b) $\beta = 2.0$

Fig. 1: IEEE 57 Bus System: Primal Infeasibility $\|F\|$ vs Time for Two Different Values of β.

When $\beta = 1.0$ (the original, non-disrupted system), the power flow problem is feasible and thus the algorithm takes the same steps as Newton's method applied to the power-flow equations. It converges *without* needing any simplex iterations; the original basis factorization is enough. As expected, the solution does not require any load shedding, and the objective is therefore zero. With $\beta = 1.2$, 1.4, or 1.8, the solution is still fully determined by the active constraints, and we observe quadratic convergence, consistently with the analysis of Subsection 3.3, though in these cases, load shedding is required. A few simplex iterations are needed in the process of solving the linear programming subproblems, to resolve the active sets. The value $\beta = 1.6$ also has a fully determined solution, but a few outer iterations are required to determine that such is indeed the case. If we do not invoke the active-set heuristic, convergence of Sℓ_1LP is slow on iterations 3 through 10, but accelerates for the final five iterations; see Figure 1a. When we use the active-set heuristics, the true active set is identified more rapidly, and fast convergence is observed at an earlier stage.

The case of $\beta = 2.0$ also shows the benefits of the active set heuristic. The Sℓ_1LP algorithm without active-set heuristic requires 34 iterations to find the solution, converging to this underdetermined solution at a linear rate, as shown in Figure 1b. When the active-set heuristic is used, the algorithm invokes it after three iterations of Sℓ_1LP and converges about four times faster as a result. We note that the initial active set estimation from the Sℓ_1LP steps was not exact for this case, but two iterations of the active-set heuristic sufficed to find the optimal active set.

When applied to these problems with β between 1 and 1.8, the MATPOWER power flow solver obtained solutions using Newton's method within ten iterations. Thus, there exist solutions to the basic power flow equations (47) without load shedding for these cases — but the voltage magnitudes at

β	Bus No.	Demanded		Injected		Loads Shed			Buses with $V = \underline{V}$
		P	Q	P	Q	%	Total P	Total Q	
1.6	20	2.3	1.0	2.02	0.88	12.4%	17.04	8.09	
	30	3.6	1.8	3.53	1.76	2.0%			20
	31	5.8	2.9	0	0	100.0%			26
	32	1.6	0.8	1.24	0.62	22.6%			34
	33	3.8	1.9	0	0	100%			53
	42	7.1	4.4	4.79	2.97	32.5%			56
	53	20	10	19.00	9.50	5.0%			57
	57	6.7	2.0	3.28	0.98	51.0%			
2.0	20	2.3	1.0	0.11	0.05	95.3%	35.62	17.16	
	25	6.3	3.2	4.90	2.49	22.3%			
	30	3.6	1.8	0.51	0.26	85.8%			20
	31	5.8	2.9	0	0	100.0%			26
	32	1.6	0.8	0.33	0.16	79.2%			35
	33	3.8	1.9	0	0	100.0%			53
	35	6.0	3.0	4.14	2.07	30.9%			56
	42	7.1	4.4	1.95	1.21	72.5%			57
	53	20	10	14.53	7.26	27.4%			
	56	7.6	2.2	6.57	1.90	13.5%			
	57	6.7	2.0	2.13	0.64	68.2%			

(P in MW / Q in MVAr)

Table 2: Feasibility Restoration Results on IEEE 57 Bus Systems with $\beta = 1.6$ and $\beta = 2.0$

some buses are too low for these solutions to be practically operational. When $\beta = 2.0$, MATPOWER fails to find a solution, suggesting strongly that the power-flow equations do not have a solution, even one with impractical voltage magnitudes.

Details of the solutions obtained by the Sℓ_1LP algorithm for the cases $\beta = 1.6$ and $\beta = 2.0$ are shown in Table 2. Load shedding is required at 8 and 11 buses, respectively, the table showing the percentage of load shedding required at each bus. The last column indicates those buses at which the optimal voltage magnitudes are at their lower bounds.

6.2 Unsolvable $N - 1$ Cases from the IEEE 300-Bus System (`case300.m`)

Another way to disrupt a power system is to remove transmission lines. We test our algorithm on the standard IEEE 300-bus system by removing each transmission line in turn — the so-called "$N - 1$" cases. Since we are mainly interested in unsolvable cases, we only consider the lines such that (a) the whole system remains fully connected when the line is removed; and (b) the Newton's method of MATPOWER fails to solve the power-flow equations for the damaged system. Using these criteria, we found fifteen unsolvable cases.

Bus			# of	Variables				Constraints		$(\underline{V},\ \overline{V})$
Ref	$\mathcal{G}$	$\mathcal{D}$	Lines	V	θ	$\sigma^{\pm}, \rho$	Total	$=$	$\leq$ or $\geq$	
1	68	231	410	231	299	227	757	530	916	(.92, 1.08)

(a) Feasibility Restoration on $N-1$ cases of IEEE 300 Bus System

Line Removed	$n-m-\lvert\mathcal{A}^*\rvert$	Iterations		Time (sec)	Objective Value	# of Buses w/ Load Sheds
		Sℓ_1LP	AS			
66	0	6(5)* 6(5)	- 1(0)	0.06 0.07	4.98e-01	2
114	0	6(1)	-	0.06	1.37e+00	1
116	Not Converged					
177	0	6(5)	-	0.06	8.15e-02	1
181	1	38(27)* 5(13)	- 5(0)	0.27 0.10	7.03e+00	3
182	0	6(3)	-	0.06	1.17e+00	2
187	Not Converged					
294	0	6(8)	-	0.06	1.54e+00	4
309	2	40(48)* 6(21)	- 8(11)	0.28 0.19	8.22e+00	7
350	Not Converged					
364	0	6(1)	-	0.06	1.08e+00	1
367	0	6(2)	-	0.06	2.29e+00	2
369	0	6(6)	-	0.06	5.10e-01	2
370	0	6(1)	-	0.06	3.48e-01	1
381	0	6(4)	-	0.06	2.33e-01	1

(b) Sℓ_1LP-AS algorithm Results on the Unsolvable $N-1$ Cases of IEEE 300-Bus System.
* indicates that the algorithm converged to an acceptable solution for which tolerance is relaxed by a factor of ten.

Table 3: $N-1$ cases of IEEE 300 Bus System

Details of the $N-1$ systems derived from the IEEE 300 Bus case are presented in Table 3a. (The original system has 411 transmission lines, but since one of them is removed in each of our cases, we list 410 lines.) Table 3b shows results for the fifteen unsolvable $N-1$ cases. As mentioned, none of these cases can be solved by the Newton's method, and three of them cannot even be solved as feasibility restoration problems, probably because of the restrictions imposed in our formulation (49). For the ten cases in which the solutions are fully determined by the constraints, we see fast convergence of Sℓ_1LP without the need to invoke the active-set heuristic. In the two underdetermined cases, use of the active-set heuristic leads to considerable speedup.

Note that in all cases, feasibility can be restored to the system by shedding load on just a few buses.

6.3 Comparisons with IPOPT and MIPS

Performance comparisons between Sℓ_1LP-AS (that is, Sℓ_1LP with the active-set heuristic) and the solvers IPOPT and MIPS for feasibility restoration problems are shown in Table 5. We used four test problems drawn from the IEEE set, with details shown in Table 4. We used a similar setup here to Subsection 6.1, modifying the basic IEEE data by scaling the impedance of all lines by a factor $\beta \geq 1$. The voltage limits on each system are chosen so that the optimal voltage magnitude values for the unmodified problems do not violate the voltage constraints.

Two different variants of IPOPT are tested. One setting uses the exact Hessian of the Lagrangian, while the other uses a limited-memory quasi-Newton (L-BFGS) approximation. MIPS, which is provided within MATPOWER, uses the exact Hessian, as does the active-set heuristic within Sℓ_1LP-AS.

The 118-bus system is quite robust. Even with $\beta = 3$, there exists a solution that can operate the system within the given voltage range with load adjustments on just a few buses. All methods solve the problem in less than .1 seconds.

On the 300-bus system, the solutions were fully determined in the cases reported. The Sℓ_1LP-AS converges rapidly without needing to invoke the active-set heuristic, and gives faster runtimes than rival approaches. We also tried setting $\beta = 1.3$, but none of the algorithms could find a solution for this case.

For the 2383-bus systems, Sℓ_1LP-AS is slightly faster than the other approaches, but its advantage over IPOPT becomes slimmer as β is increased. On these heavily disrupted systems, many buses require adjustments to recover feasibility, and the solutions are underdetermined, so the dual simplex algorithm requires many more iterations on each subproblem. The active-set heuristic continues to work well, however, and yields fast local convergence.

Sℓ_1LP-AS is significantly faster for the cases based on the 2476-bus system, even though there are underdetermined solutions in some of these cases.

We conclude with some observations about the implementations of each of these solvers. Note that both IPOPT (with exact Hessian) and MIPS both use interior point approaches and require similar numbers of iterations, but IPOPT is about 30% faster on the more difficult problems, probably because it is implemented in MATLAB whereas IPOPT is implemented in `C++`. Our Sℓ_1LP-AS algorithm is coded in a combination of MATLAB (to set up the subproblems and to perform the active-set heuristic) and native code (the CPLEX solver for the LP subproblems) and thus falls somewhere between IPOPT and MIPS in the efficiency of its code. A fully native implementation of Sℓ_1LP-AS has the potential to perform even better.

7 Conclusions

We have proposed an algorithm consisting of Sℓ_1LP and an active-set heuristic to solve nonsmooth penalty-function formulations of nonlinear programming

System	Buses	Lines	Variables	Constraints		$(\underline{V}, \overline{V})$
				$=$	$\leq$ or $\geq$	
118 Bus	118	186	262	181	290	(0.93, 1.07)
300 Bus	300	411	757	530	916	(0.92, 1.08)
2383 Bus	2383	2896	6087	4438	7410	(0.90, 1.12)
2746 Bus	2746	3279	6925	5127	8360	(0.98, 1.20)

Table 4: Power System Information

System	β	Sℓ_1LP-AS			IPOPT				MIPS		# of Active Constraints	# of Buses with Loads Shed
		Iterations		Time (s)	L-BFGS		Hessian		Iter.	Time (s)		
		LP	AS		Iter.	Time (s)	Iter.	Time (s)				
118 Bus	1.5	4(0)	-	0.04	6	0.05	7	0.04	12	0.06	262	0
	2.0	5(2)	-	0.04	9	0.07	10	0.05	13	0.06	262	2
	2.5	5(17)	3(0)	0.05	14	0.09	16	0.08	16	0.08	261	8
	3.0	4(15)	3(1)	0.05	13	0.09	15	0.07	15	0.07	261	14
300 Bus	1.1	6(5)	-	0.06	15	0.19	16	0.15	18	0.20	757	5
	1.2	6(11)	-	0.05	16	0.20	18	0.16	18	0.18	757	11
2383 Bus	1.2	7(22)	1(0)	0.29	22	1.54	20	1.14	22	1.85	6087	17
	1.4	5(36)	5(3)	0.58	24	1.78	23	1.32	25	2.09	6086	22
	1.6	8(91)	5(2)	0.68	32	2.47	21	1.18	22	1.84	6083	31
	1.8	15(542)	5(5)	1.27	107	8.62	31	1.74	33	2.09	6068	72
	2.0	17(913)	5(3)	1.40	120	9.51	32	1.79	29	2.40	6052	102
2746 Bus	1.2	5(1)	1(0)	0.23	12	1.01	14	0.95	16	1.45	6925	1
	1.4	5(5)	1(0)	0.23	21	1.75	22	1.46	19	1.69	6925	7
	1.6	4(15)	4(1)	0.47	40	2.98	29	1.87	27	2.49	6924	22
	1.8	5(27)	4(4)	0.56	40	3.30	30	1.94	31	2.78	6923	36
	2.0	7(130)	4(4)	0.79	93	8.58	33	2.13	34	3.08	6916	54

Table 5: Performance Comparisons: Sℓ_1LP (with Active-Set Heuristic) vs IPOPT and MIPS

problems. Global and local convergence properties were explored, with a focus on local (superlinear) convergence in the case in which the solution is fully determined by the constraints — a situation that occurs frequently in our target application. In the final sections, we described application of our algorithm to the problem of restoring feasibility to a disrupted power system, in a practical way that limits the number of buses at which load must be shed. Computational results were presented for system sizes up to 2746 buses, and comparisons were performed with interior-point solvers on the same formulations.

Since the Sℓ_1LP algorithm needs to solve LP subproblems at each iteration, it may become slow due to the overhead of finding the LP solutions for large problems. For highly perturbed power systems that require many demand buses to be adjusted to recover feasibility, many simplex iterations may be required to find a subproblem solution. Because of the nonconvex formulation, our approach is not guaranteed to find a global solution, a limitation shared with other solvers, including interior-point methods.

One goal of our formulation and method is to guide system operators toward load-shedding patterns that restore practical operation of the grid at minimum disruption. Another goal is to use the optimal objective in (49) as a measure of disruption to the grid, to be used in analyzing the vulnerability of the grid to deliberate attacks or natural disturbances. Vulnerability analysis may indicate what capital improvements could make the system more robust to such disruptions. We are exploring these issues further in current research.

A Solving Augmented Linear Systems

Assume that we already know the solution x_0 of the square linear system

$$Hx = b_1,$$

where H is symmetric, as well as factors L (lower triangular) and D (block diagonal) such that $LDL^T = H$. Suppose we are presented with the following augmented square linear system:

$$\begin{bmatrix} H & V \\ X & S \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

where the dimensions of the square matrix S are much smaller than those of H. We first factorize the matrix into the two block triangular matrices:

$$\begin{bmatrix} H & V \\ X & S \end{bmatrix} = \begin{bmatrix} H & 0 \\ X & C \end{bmatrix} \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix}$$

where the matrices C and Y can be obtained by solving the following problems:

$$\begin{aligned} V = HY \quad &\Rightarrow \quad LDL^TY = V, \\ S = XY + C \quad &\Rightarrow \quad C = S - XY. \end{aligned}$$

Because of the properties of L and D, the matrix Y can be calculated economically, while C requires simply a matrix multiplication. We can therefore rewrite the augmented system

as two linear systems with auxiliary variables w_1 and w_2:

$$\begin{bmatrix} H & \\ X & C \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \qquad \begin{bmatrix} I & Y \\ & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

Since $Hw_1 = b_1$, we have $w_1 = x_0$. Then w_2 is obtained by solving the following system:

$$Cw_2 = b_2 - Xw_1,$$

which can be performed economically, since C is small. We can then find x_1 and x_2 by setting

$$\begin{aligned} x_2 &= w_2, \\ x_1 &= w_1 - Yx_2. \end{aligned}$$

References

Barboza L, Salgado R (2001a) Restoring solutions for unsolvable cases via minimum load shedding for a specified direction. In: 22nd International Conference on Power Industry Computer Applications (PICA) 2001, Sydney, NSW, pp 374–379

Barboza L, Salgado R (2001b) Unsolvable power flow analysis-an approach based on interior point nonlinear optimization methods. In: IEEE Power Tech Proceedings 2001, Porto, Portugal, vol 2

Fletcher R (1987) Practical Methods of Optimization, 2nd edn. John Wiley and Sons, New York

Fletcher R, Sainz de la Maza E (1989) Nonlinear programming and nonsmooth optimization by successive linear programming. Mathematical Programming 43:235–256

Granville S, Mello J, Melo A (1996) Application of Interior point methods to power flow unsolvability. IEEE Transactions on Power Systems 11(2):1096–1103

Hare W, Lewis A (2004) Identifying active constraints via partial smoothness and prox-regularity. Journal of Convex Analysis 11(2):251–266

Iwamoto S, Tamura Y (1981) A load flow calculation method for ill-conditioned power systems. IEEE Transactions on Power Apparatus and Systems PAS-100(4):1736–1743

Lavaei J, Low SH (2012) Zero duality gap in optimal power flow problem. IEEE Transactions on Power Systems 27(1):92–107

Lewis A (2003) Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization 13:702–725

Milano F (2009) Continuous Newton's method for power flow analysis. IEEE Transactions on Power Systems 24(1):50–57

Min W, Shengsong L (2005) A trust region interior point algorithm for optimal power flow problems. International Journal of Electrical Power & Energy Systems 27(4):293–300

Molzahn DK, Lesieutre BC, DeMarco CL (2013) A sufficient condition for power flow insolvability with applications to voltage stability margins. IEEE Transactions on Power Systems 28(3):2592–2601

Nocedal J, Wright SJ (2006) Numerical Optimization, 2nd edn. Springer, New York

Oberlin C, Wright SJ (2006) Active set identification in nonlinear programming. SIAM Journal on Optimization 17(2):577–605

Overbye TJ (1995) Computation of a practical method to restore power flow solvability. IEEE Transactions on Power Systems 10(1):280–287

Rockafellar RT (1970) Convex Analysis. Princeton University Press, Princeton, N.J.

Rockafellar RT, Wets RJB (1998) Variational analysis. Springer

Sousa AA, Torres GL (2007) Globally convergent optimal power flow by trust-region interior-point methods. In: IEEE Power Tech 2007, Lausanne, Switzerland, pp 1386–1391

Sousa AA, Torres G, Cañizares CA (2011) Robust optimal power flow solution using trust region and interior-point methods. IEEE Transactions on Power Systems 26(2):487–499

Tripathy SC, Prasad GD, Malik OP, Hope GS (1982) Load-flow solutions for ill-conditioned power systems by a Newton-like method. IEEE Transactions on Power Apparatus and Systems PAS-101(10):3648–3657

University of Washington, Electrical Engineering (2014) Power Systems Test Case Archive. URL http://www.ee.washington.edu/research/pstca/

Vanderbei R (2007) Linear Programming: Foundations and Extensions. Springer

Wächter A, Biegler L (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106(1):25–57

Wright SJ (1993) Identifiable surfaces in constrained optimization. SIAM Journal on Control and Optimization 31:1063–1079

Yildirim EA, Wright SJ (2002) Warm-start strategies in interior-point methods for linear programming. SIAM Journal on Optimization 12:782–810

Zhou R, Zhang Y, Yang H (2005) A trust-region algorithm based on global SQP for reactive power optimization. In: 2nd International Conference on Electrical and Electronics Engineering 2005, pp 292–295

Zimmerman RD, Murillo-Sánchez CE, Thomas RJ (2011) MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on Power Systems 26(1):12–19