

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

A Filter Active-Set Trust-Region Method

Michael P. Friedlander, Nick I. M. Gould, Sven Leyffer, and Todd S. Munson

Mathematics and Computer Science Division

Preprint ANL/MCS-P1456-0907

September 10, 2007

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Contents

1	Introduction	1
2	The Algorithm	3
2.1	Sequential Linear-Quadratic Programming Methods	3
2.2	Difficulties with the LP/TR Step Computation	4
2.3	An Alternative Step Strategy via Regularized LP	5
2.4	Globalization and the Filter	6
2.5	Feasibility Restoration	8
2.6	Adjusting the Regularization Parameter μ	8
2.7	Computation of Multipliers	10
2.8	Algorithm Statement	10
3	Convergence Analysis	12
3.1	Preliminary Lemmas	12
3.2	Main Convergence Result	17
4	Properties of Trust-Region Subproblem	19
4.1	Active Set Identification Properties of RLP	19
4.2	Relationship between TR Parameters ρ and μ	21
5	Conclusions	21

A Filter Active-Set Trust-Region Method*

MICHAEL P. FRIEDLANDER,[†]NICK I. M. GOULD,[‡]
SVEN LEYFFER, AND TODD S. MUNSON[§]

September 10, 2007

Abstract

We develop a new active-set method for nonlinear programming problems that solves a regularized linear program to predict the active set and then fixes the active constraints to solve an equality-constrained quadratic program for fast convergence. Global convergence is promoted through the use of a filter. We show that the regularization parameter fulfills the same role as a trust-region parameter, and we give global convergence results. In addition, we show that the method identifies the optimal active set once it is sufficiently close to a regular solution. We also comment on alternative regularized problems that allow the inclusion of curvature information into the active-set identification.

Keywords: Nonlinear programming, active-set methods, filter methods.

AMS-MSC2000: 90C30, 90C33, 90C51, 49M37, 65K10.

1 Introduction

Consider the nonlinear program (NLP)

$$(P) \begin{cases} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c_i(x) \geq 0, \quad i = 1, \dots, m, \end{cases}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are twice continuously differentiable. NLPs have a broad range of applications in engineering, science, and economics.

We are interested in developing fast active-set methods for large-scale NLPs. Current active-set methods solve a quadratic program (QP) or a linear program (LP) to predict the active set. This step is computationally expensive, however, and does not readily generalize to large problems, because the LP or QP subproblems are solved with pivoting algorithms that can change only one activity per iteration. In addition, active-set quadratic programming methods must factorize either a dense reduced Hessian matrix (in the case of

*Preprint ANL/MCS-P1456-0907

[†]Department of Computer Science, University of British Columbia, Vancouver, Canada, mpf@cs.ubc.ca

[‡]University of Oxford, England, UK, nick.gould@comlab.ox.ac.uk

[§]Mathematics and Computer Science Division, Argonne National Laboratory, {leyffer,tmunson}@mcs.anl.gov

a null-space method) or a dense Schur-complement (in the case of a range-space method). We therefore believe that sequential quadratic programming approaches are not suitable for large-scale NLPs unless either the null-space or the range-space is small.

The bottleneck of current active-set methods for NLP is the determination of the active set. Our strategy is to replace this expensive computational step with a cheaper alternative. In particular, we consider gradient projection methods that allow much larger changes in the active set at each inner iteration. By using computational kernels for which good parallel solvers exist, we also ensure that the proposed method can be implemented in parallel. In this paper, we explore the theoretical underpinnings of the new method.

But why should we look at active-set methods at all, given the recent surge of interior-point methods and their clear computational superiority? The answer to this question lies in two classes of optimization problems that have received recent attention, namely, mixed-integer nonlinear programming (MINLP) and optimization problems with partial differential equation (PDE) constraints. Methods for solving MINLP solve NLPs as subproblems, usually starting from an excellent guess of the optimal solution. Interior-point methods are not readily warm-started and are not as efficiently warm-started as are active-set methods, despite recent efforts in the area [2, 11, 13].

PDE-constrained optimization problems give rise to huge linear systems that must be solved at every iteration. In particular, the linear systems arising from any optimization method can no longer be solved by using direct solvers or factorizations. Instead, iterative solvers must be employed. This shift has important implications: pivoting methods for LP or QP subproblems rely on the fact the matrix factorizations can be updated cheaply after each pivot. Clearly, this approach is no longer possible if the linear systems are solved iteratively. The reliance of interior-point methods on factorizations is less obvious. By factorizing the linear systems, interior-point methods avoid the inherent ill-conditioning due to the barrier parameter. Thus, to solve PDE-constrained optimization problems, we must develop methods that use computational kernels that can be solved efficiently with iterative solvers and whose conditioning is not artificially inflated.

Our new active-set method solves a regularized LP (RLP) to predict the active set and then fixes the active constraints and solves an equality-constrained QP (EQP) to speed convergence. We show below that the dual of the regularized LP is a bound-constrained QP for which efficient solution methods such as projected-gradient methods exist. A simple outline of our algorithm is given in Figure 1. This figure, however, leaves a number of important open questions: When should we accept a step? What happens if the RLP or the EQP have no solution? In a practical implementation we might also restrict the EQP step by a trust-region or a proximal-point term, and we can either use the RLP step if the EQP has no solution, or use a piecewise line-search along an arc.

This paper is organized as follows. In the next section, we present our new algorithm, FASTr, for solving NLP. In Section 3 we show that FASTr is globally convergent. Section 4 contains preliminary results regarding the active-set identification property of our algorithm. Section 5 outlines what is left to be done.

We finish this section by briefly introducing our notation. We denote the gradient of the objective function by $g(x) = \nabla f(x)$, the constraints by $c(x) = (c_1(x), \dots, c_m(x))^T$, and the Jacobian by $A(x) = \nabla c(x)^T$ with $a_i(x) = \nabla c_i(x)$. We also use subscripts to denote functions and gradients evaluated at iterates, for example, $c_k = c(x_k)$.

```

Given an initial point  $x_0$ , set  $k = 0$ , and  $\mu_0 > 0$ .
while  $d \neq 0$  do
  Solve the regularized LP, RLP( $x_k, \mu_k$ ) for step  $d_k$ 
  Identify the active set  $\mathcal{A}(x_k + d_k)$ .
  Solve the equality constrained QP, EQP( $x + d$ ) for a second-order step  $d_{qp}$ .
  if  $x_k + d_{qp}$  acceptable step then
    | Set  $x_{k+1} := x_k + d_{qp}$ , and possibly increase  $\mu_{k+1} = 2\mu_k$ .
  else
    | Set  $x_{k+1} := x_k$ , and decrease  $\mu_{k+1} = \mu_k / 2$ .
  end
end

```

Figure 1: RLP-EQP algorithm outline

2 The Algorithm

We start by reviewing sequential linear-quadratic programming (SLQP) methods and derive an alternative subproblem to identify the optimal active set. Next, we describe our globalization strategy and provide a formal statement of our new algorithm.

2.1 Sequential Linear-Quadratic Programming Methods

We believe SLQP methods [3, 4, 6, 24] are well suited for large-scale nonlinear programming, simply because each of the key steps involved—the solution an LP or EQP, together with its attendant linear algebra—has already been tuned to cope with big problems. The basic idea behind such methods is simple: in order to improve an estimate x of a solution to NLP, NLP is replaced by an LP linearization about x of the form

$$(\text{LP}(x)) \begin{cases} \underset{d}{\text{minimize}} & g^T(x)d \\ \text{subject to} & c(x) + A^T(x)d \geq 0. \end{cases}$$

Since the solution to the resulting LP—if there is one—is most likely at a (possibly distant) vertex (or possibly at infinity), the LP solution d is further constrained so that $\|d\| \leq \rho$ for some suitable norm $\|\cdot\|$ and positive “radius” ρ , leading to the trust-region subproblem

$$(\text{TR}(x, \rho)) \begin{cases} \underset{d}{\text{minimize}} & g^T(x)d \\ \text{subject to} & c(x) + A^T(x)d \geq 0 \\ \text{and} & \|d\| \leq \rho. \end{cases}$$

The intention is then that distant constraint linearizations in $\text{TR}(x, \rho)$ play no role, and the hope is that the closer ones give a good prediction of the optimal active set for NLP.

In practice, the resulting step d may not necessarily be particularly effective, since it is a first-order (“constrained steepest descent”) direction. But the set of predicted active constraints,

$$\mathcal{A} := \mathcal{A}(x) := \{i | c_i(x) + a_i(x)^T d = 0\},$$

can be used to generate a more appropriate subproblem. In particular, given such a prediction, those constraints that are inactive may be temporarily discarded, and a more effective (second-order) step computed by approximately solving an EQP in which a model of the Lagrangian function is minimized subject to the linearized constraints in \mathcal{A} being held as equalities, that is,

$$(\text{EQP}(x, \mathcal{A})) \begin{cases} \underset{d}{\text{minimize}} & \frac{1}{2}d^T H d + g^T(x)d \\ \text{subject to} & A_{\mathcal{A}}^T(x)d = -c_{\mathcal{A}}(x), \end{cases}$$

where $c_{\mathcal{A}}(x) = (c_i(x))_{i \in \mathcal{A}}$, $A_{\mathcal{A}}(x)$ are the columns of the Jacobian $A(x)$ corresponding to active constraints \mathcal{A} , and $H \approx \nabla^2 L(x, y)$ approximates the Hessian of the Lagrangian, $L(x, y) = f(x) - y^T c(x)$, for NLP for some suitable Lagrange multiplier estimates y . Since the solution to this problem may be unbounded from below, it is common to introduce an additional ellipsoidal trust-region constraint $\|d + d_c\|_G \leq \rho_2$ (for G positive definite, and $\|z\|_G = \sqrt{z^T G z}$) to ensure that $\text{EQP}(x, \mathcal{A})$ has a solution, where d_c is the minimum norm solution to $A_{\mathcal{A}}^T(x)d = -c_{\mathcal{A}}(x)$.

Many other precautions are necessary in practice. These include (i) the means to recover from inconsistent constraint linearizations and incompatibilities caused by small ρ , (ii) a mechanism to assess the quality of EQP step, (iii) the possible need for a second-order correction step to avoid the Maratos effect, and (iv) a strategy for adjusting the step restrictions to try to encourage accurate active-set identification and fast convergence.

2.2 Difficulties with the LP/TR Step Computation

From a pragmatic point of view, it has been traditional [3, 4, 6, 24] to specify a polyhedral trust-region norm when solving $\text{TR}(x, \rho)$, since then the subproblem may be reformulated and solved as an LP. This has an unfortunate side effect, however. Specifically, while it may be reasonable to hope that the active set of the linearized constraints for $\text{TR}(x, \rho)$ approaches the optimal active set for NLP as x approaches a solution to NLP, there is no reason to suppose the same is true for those defining the polyhedral trust region. Indeed, simply reducing ρ for fixed x can dramatically alter the active trust-region constraints. This is unfortunate because much of the perceived efficiency of SLP/SQP methods derives from the hope that the active set for one subproblem is a good prediction for that of the next (and thus allows efficient “warm starts”). Extensive numerical experience with the *SLIQUE* and *KNITRO-ACTIVE* software packages [3, 5] has indicated that indeed in some cases warm-start strategies may be inefficient for this reason.

One way to avoid this drawback is to use a trust-region norm that does not have multiple faces. Then, even if the trust-region constraint is active, it simply defines a single face and thus does not add significantly to the combinatorial complexity. Any ℓ_p trust-region norm, $p \in (1, \infty)$, is appropriate, but the downside is that $\text{TR}(x, \rho)$ is no longer an LP and may be more difficult to solve—there has been some work for the ℓ_2 -norm case using semi-definite and cone programming techniques [1, 12, 17, 18, 20, 26], but to our knowledge the size of problems that are amenable to such techniques fall far short of those currently possible for LP subproblems.

A second difficulty, common to many trust-region methods for constrained optimization, is that $\text{TR}(x, \rho)$ has no solution for any sufficient small ρ whenever c is nonzero. This is

avoided by many modern SLQP methods [3, 24] by replacing $\text{TR}(x, \rho)$ by

$$\underset{d}{\text{minimize}} \quad g^T(x)d + \pi \|(\max(0, -c(x) - A^T(x)d))\| \quad \text{subject to} \quad \|d\| \leq \rho$$

for some positive penalty parameter π , but issues then arise [4] on precisely how to choose π .

2.3 An Alternative Step Strategy via Regularized LP

Our interest then is to develop SLQP-type methods that do not suffer from the aforementioned defects. Instead of using $\text{TR}(x, \rho)$, we propose an alternative trust-region subproblem that can be viewed as adding a penalizing term for the trust-region constraint $\|d\|_2 \leq \rho$ to the objective. We shall refer to this as a *regularized LP* (RLP):

$$(\text{RLP}(x, \mu)) \begin{cases} \underset{d}{\text{minimize}} & \mu g^T(x)d + \frac{1}{2}d^T d \\ \text{subject to} & c(x) + A^T(x)d \geq 0. \end{cases}$$

Later, we will show a close relationship between $\text{RLP}(x, \mu)$ and $\text{TR}(x, \rho)$. Crucially, we control the convergence of our method by adjusting μ just as a trust-region SLQP method does by adjusting ρ . Of course $\text{RLP}(x, \mu)$ is actually a (parametrized) convex quadratic program (QP), and thus a wealth of suitable software exists for solving it (see [14, §4.2]). We need to be cautious, however, since $\{d : c(x) + A^T(x)d \geq 0\}$ may be empty. We return to this issue below, but assume for now that one can find a solution d_{LP} to $\text{RLP}(x, \mu)$.

The problem $\text{RLP}(x, \mu)$ has an interesting history. In particular, $\text{RLP}(x, \mu)$ is a Tikhonov regularization [27] of $\text{LP}(x)$, for which the solution of the former is the least ℓ_2 -norm solution to the latter for all μ sufficiently large [22]. See [29] for further references. Vitaly, since the Hessian of the objective is diagonal, the dual of $\text{RLP}(x, \mu)$ is a bound-constrained quadratic program (BQP) [16, 21] in the dual variables y ,

$$(\text{BQP}(x, \mu)) \begin{cases} \underset{y}{\text{minimize}} & \frac{1}{2}y^T A^T(x)A(x)y + (c(x) - \mu A^T(x)g(x))^T y + \frac{\mu^2}{2}g^T(x)g(x) \\ \text{subject to} & y \geq 0, \end{cases}$$

from which the solution $d = -\mu g(x) + A(x)y$ of $\text{RLP}(x, \mu)$ may be recovered. The significance is that there are excellent methods [7, 15, 21, 23] and software [8, 19] for solving large-scale BQPs.

Having solved $\text{RLP}(x, \mu)$ —or its dual, $\text{BQP}(x, \mu)$ —to find an initial step d_{LP} and its associated active set \mathcal{A} , our algorithm solves the resulting $\text{EQP}(x, \mathcal{A})$ to find a second step, d_{QP} (with, as before, an additional trust-region constraint as a precaution). So long as H is positive definite on the null-space of $A_{\mathcal{A}}^T$, we can find d_{QP} by solving the linear system arising from the first-order optimality conditions of $\text{EQP}(x, \mathcal{A})$,

$$\begin{pmatrix} H & A_{\mathcal{A}}(x) \\ A_{\mathcal{A}}^T(x) & 0 \end{pmatrix} \begin{pmatrix} d_{QP} \\ -y_{\mathcal{A}}(x) \end{pmatrix} = \begin{pmatrix} -g \\ -c_{\mathcal{A}}(x) \end{pmatrix},$$

although this may not necessarily satisfy the additional trust-region constraint. Better, perhaps, is to use a projected, preconditioned conjugate-gradient method (see [14, §4.1]) to

find d_{QP} , since this automatically takes into account the elliptical trust-region constraint and is highly suited to large-scale EQPs. We may also wish to compute some form of second-order correction (SOC) d_{SOC} (see [9, §15.3.2.3]) to account for constraint nonlinearity when the constraint residuals are small.

We thus have two, perhaps three, potential steps. The RLP step d_{LP} can be used to ensure global convergence, and so we will fall back on it whenever d_{QP} and d_{SOC} are unsuitable. The EQP step d_{QP} and the SOC step d_{SOC} are intended to accelerate convergence and thus to be preferred if suitable. We also note that the computation of superficially similar steps has recently been proposed by Yamashita and Dan [28] but with significant differences, the most important being that μ is not used as a control parameter as it is here and that a different globalization strategy is used.

2.4 Globalization and the Filter

Turning to globalization, we denote the *predicted reduction* in the linear model of $f(x)$ due to d_{LP} , if the latter exists, by

$$\Delta l = -g^T(x)d_{LP}$$

and let

$$\Delta f = f(x) - f(x + d)$$

be the *actual reduction* in $f(x)$ resulting from a given step d . We measure constraint infeasibility by

$$h(c(x)) = \|c(x)^+\|_\infty,$$

where $c_i(x)^+ = \max(0, -c_i(x))$.

We now define a *filter* [10] that combines the two competing aims in NLP, namely, minimization of $f(x)$ and $h(c(x))$. We will consider pairs (h, f) obtained by evaluating $f(x)$ and $h(c(x))$. We say that a pair (h_k, f_k) dominates another pair (h_l, f_l) if both $h_k \leq h_l$ and $f_k \leq f_l$. A filter \mathcal{F} is a collection of pairs (h, f) such that no pair dominates another pair.

We say that a point is *acceptable* to a filter \mathcal{F} if its (h, f) pair satisfies

$$h \leq \beta h_i \text{ or } f \leq f_i - \gamma h, \quad \forall i \in \mathcal{F}, \quad (2.1)$$

where β and γ are constants such that $0 < \gamma < \beta < 1$. The first inequality corresponds to sufficient reduction in h , while the second inequality follows [6]. Both inequalities create an envelope around the filter, which prevents convergence to a point at which $h > 0$. A typical filter with three entries is indicated in Figure 2, where the solid lines indicate the filter and the dotted lines show the envelope. In practice, we also add an upper bound

$$h(c(x)) \leq \beta u \quad (2.2)$$

on the constraint violation to the filter. We will use acceptability to the filter as a means to assess potential new iterates $x + d$; in particular, for $x + d$ to be considered as the new iterate, it must be acceptable for the augmented filter $\mathcal{F} \cup (h(c(x)), f(x))$.

This definition of a filter is not sufficient to ensure convergence because the iterates might still converge to a feasible, but noncritical point. To avoid this situation we insist

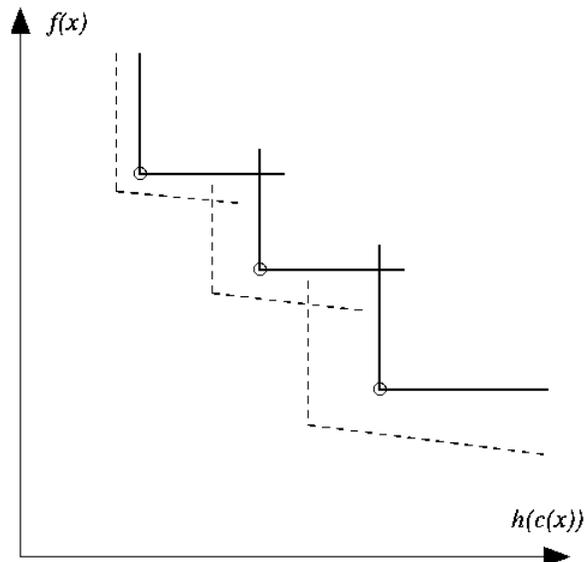


Figure 2: A typical NLP filter and its envelope

on a sufficient decrease condition whenever the linear model predicts decrease. Thus, for $\sigma \in [0, 1)$ we ask that

$$\Delta f \geq \sigma \Delta l \quad (2.3)$$

whenever

$$\Delta l \geq \delta h^2, \quad (2.4)$$

where $\delta > 0$. We also define the *Cauchy step*

$$d_C = \alpha_C d_{LP}$$

as the step along d_{LP} at which the decrease in the quadratic model

$$\Delta q(d) = -d^T g(x) - \frac{1}{2} d^T H d$$

is maximized for $0 < \alpha_C \leq 1$. An alternative to the above sufficient reduction condition is then to ask for

$$\Delta f \geq \sigma \Delta q(d_C);$$

it is straightforward to show [6, Lemma 3] that $\Delta q(d_C) \geq \frac{1}{2} \alpha_C \Delta l$. A successful iteration for which (2.4) does not hold is referred to as an h-type iteration. A successful iteration at which both (2.4) and (2.3), or $\Delta f \geq \sigma \Delta q(d_C)$ hold is referred to as an f-type iteration. Broadly speaking, an h-type iteration mainly reduces the constraint violation, which an f-type iteration also makes progress at reducing the objective function.

We note that an alternative is to ask for sufficient reduction whenever $\Delta l > 0$. The choice of sufficient reduction condition is directly related to the way the filter is updated. If we use (2.3) and (2.4), then we add acceptable points to the filter only for which this condition fails (i.e. the so-called h-type steps), because (2.3) and (2.4) ensure that $h \rightarrow 0$ on so-called

f-type iterations that are not added to the filter. On the other hand, if we use (2.3) and $\Delta l > 0$, then we must add all points to the filter for which $h > 0$ holds. We prefer to use (2.4), as this potentially adds fewer points to the filter than does the condition $\Delta l > 0$.

2.5 Feasibility Restoration

As we have noted, $\text{RLP}(x, \mu)$ has no solution if $\mathcal{L}(x) = \{d : c(x) + A^T(x)d \geq 0\}$ is empty. In this case, we temporarily abandon any attempt to reduce $f(x)$ and instead enter a *restoration* phase [10] with the sole aim of reducing the infeasibility $h(c(x))$. Since x has proved not to be a satisfactory point for the algorithm, we first add its $(h(c(x)), f(x))$ pair to the filter—this may mean that certain current filter points are now dominated, and these can safely be removed (we call this *tidying up* the filter). We may then apply any algorithm we like to perform our restoration, but our aim is to find a new point \bar{x} that is acceptable to the filter and *compatible* for the resulting $\text{RLP}(\bar{x}, \mu)$ in the sense that $\mathcal{L}(\bar{x}) \neq \emptyset$.

We also enter a restoration phase whenever there is evidence that reducing μ further will not change the solution to $\text{RLP}(x, \mu)$ significantly. The next subsection presents one possible strategy for detecting this situation.

Various algorithms exist for the restoration phase. We note that it is always possible to generate a filter-acceptable point if the NLP is feasible, because the smallest constraint violation of any filter entry is positive:

$$\tau_k = \min_{i \in \mathcal{F}^k} h_i > 0. \quad (2.5)$$

However, the restoration phase may also converge to stationary point of constraint violation, which we take as an indication that the NLP is inconsistent.

2.6 Adjusting the Regularization Parameter μ

It is not surprising that $\text{RLP}(x, \mu)$ and $\text{TR}(x, \rho)$ are closely related. In fact, we show below that ρ is a piecewise quadratic function of μ . In Figure 3, we illustrate how the trust-region radius ρ and the regularization parameter μ are related. The left figure shows the solution path of the RLP

$$\begin{aligned} & \underset{d}{\text{minimize}} && \mu d_1 + \frac{1}{2}(d_1^2 + d_2^2) \\ & \text{subject to} && d_1 + d_2 \geq 3 \\ & && -d_1 + d_2 \geq -2 \\ & && 1 \leq d_2 \leq 3, \end{aligned}$$

from $\mu = 8$ following the pink line to $\mu = 0$, which corresponds to the projection onto the feasible set. The figure on the right shows the piecewise quadratic relationship between ρ and μ . The flat regions in this figure correspond to ranges of μ for which the solution d of RLP is constant, namely, the corners of the solution path.

The close relationship between $\text{RLP}(x, \mu)$ and $\text{TR}(x, \rho)$ is further examined below and motivates the adjustment of the regularization parameter μ in much the same way in which ρ is adjusted. However, some care has to be taken when adjusting μ to avoid inefficiencies, because the RLP step d and μ are only indirectly linked. For example, standard trust-region

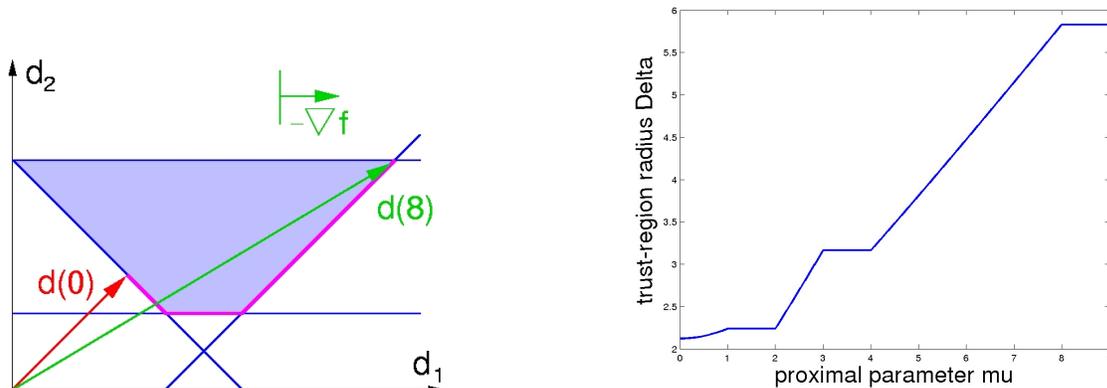


Figure 3: Relationship between perturbation parameter μ and trust-region radius ρ .

methods adjust ρ using the rule

$$\rho_{k+1} = \min(\|d_k\|, \rho_k)/2,$$

to ensure that the next trust-region subproblem is significantly different. Clearly, a similar rule is not possible for algorithms using RLP.

To reduce μ , we perform a parametric search to find the breakpoints of $\text{RLP}(x, \mu)$. The aim is to find the largest $\Delta\mu$ such that $\text{RLP}(x, \mu^+)$, where $\mu^+ = \mu - \Delta\mu$ still has the same active set. Thus, for a given active set (with Jacobian submatrix A), we have

$$\begin{pmatrix} d(\mu^+) \\ y(\mu^+) \end{pmatrix} = \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} -\mu^+ g \\ -c \end{pmatrix} = \begin{pmatrix} d(\mu) \\ y(\mu) \end{pmatrix} + \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} \Delta\mu g \\ 0 \end{pmatrix}.$$

Defining

$$\begin{pmatrix} \Delta d \\ \Delta y \end{pmatrix} := \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} g \\ 0 \end{pmatrix},$$

we have that

$$\begin{pmatrix} d(\mu^+) \\ y(\mu^+) \end{pmatrix} = \begin{pmatrix} d(\mu) \\ y(\mu) \end{pmatrix} + \Delta\mu \begin{pmatrix} \Delta d \\ \Delta y \end{pmatrix}.$$

This gives one bound on the maximum value of $\Delta\mu$. In addition, we must ensure that the step also remains feasible with respect to the inactive constraints $i \in \mathcal{I}$. Combining these two conditions gives the following ratio test to ensure that the active set remains unchanged:

$$\Delta\mu = \begin{cases} \operatorname{argmax}_{\Delta\mu \geq 0} & \Delta\mu \\ \text{subject to} & y(\mu) + \Delta\mu \Delta y \geq 0 \\ & c_i + a_i^T (d(\mu) + \Delta\mu \Delta d) \geq 0 \quad \forall i \in \mathcal{I}. \end{cases} \quad (2.6)$$

We note that if $\mu^+ = 0$ in (2.6), then we should enter the restoration phase, because reducing μ is unlikely to change the RLP step to generate an acceptable point. Otherwise, we set $\mu^+ = (\mu - \Delta\mu)/2$.

If the step is acceptable, then we increase μ to be within a fixed range $[\mu, \bar{\mu}]$, where $\bar{\mu}$ can be chosen to allow the usual doubling of the regularization parameter if we observe good agreement between predicted and actual reduction.

2.7 Computation of Multipliers

The solution of the EQPs requires multiplier estimates that are used in the (approximation) of the Hessian of the Lagrangian, H . These multiplier estimates may be the multipliers from the previous EQP step. Alternatively, if the previous step was an RLP step, we can compute first-order multiplier estimates by solving the following system:

$$\begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = - \begin{pmatrix} g \\ c \end{pmatrix}.$$

As part of the solution of $\text{RLP}(x, \mu)$, we have already solved the following two systems:

$$\begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} x_r \\ y_r \end{pmatrix} = - \begin{pmatrix} \mu g \\ c \end{pmatrix} \quad \text{and} \quad \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix} = - \begin{pmatrix} g \\ 0 \end{pmatrix}.$$

Thus, first-order multiplier estimates can be obtained as a suitable linear combination of these two solutions, namely,

$$y = y_r - (\mu - 1)y_p.$$

2.8 Algorithm Statement

Our algorithm is known as the filter active-set trust-region method, or FASTr for short (see page 11); of course, only thorough numerical experiments can show whether this name is justified. The algorithm consists of two nested loops. In the inner loop, the regularization parameter μ is reduced until either an acceptable point is found or we enter the restoration phase. For the purposes of this proof, we count the inner loop as one iteration.

We show that FASTr is globally convergent in the next section.

```

Given  $(x_0, y_0)$ ,  $\mu_0$ , and upper bound  $U$ , set  $k \leftarrow 0$ ; compute  $\nabla f_k, \nabla c_k$ 
while not optimal do
  reset regularization parameter  $\mu \in [\underline{\mu}, \bar{\mu}]$ 
  repeat
    solve RLP( $x_k, \mu$ ) or its dual for the first-order step  $d_{LP}$ 
    if  $\exists$  solution  $d_{LP}$  of LP( $x_k, \mu$ ) &  $\Delta\mu > 0$  in (2.6) then
      if  $d_{LP} = 0$  then terminate KKT point found
      compute predicted linear reduction  $\Delta l$ 
      evaluate Hessian  $H \approx \nabla^2 L(x_k, y_k)$  & solve EQP( $x_k, \mathcal{A}$ ) for step  $d_{QP}$ 
      compute the Cauchy-step  $d_C, \alpha_C$  and  $\Delta q_C$ 
      success  $\leftarrow$  false
      for  $d \in \{d_{QP}, d_{LP}\}$  do
        evaluate  $f(x_k + d)$  and  $h(c(x_k + d))$ 
        if  $x_k + d$  acceptable to filter and  $(h_k, f_k)$  then
          if  $\Delta f \geq \sigma \min(\Delta q_C, \Delta l)$  or  $\Delta l < \delta(h_k)^2$  then
            success  $\leftarrow$  true
            exit for loop
          end
        end
      end
      if success = true then
        set  $\mu_k = \mu, d_k = d, \Delta l_k = \Delta q, \Delta f_k = \Delta f$ 
        if  $\Delta l_k < \delta(h_k)^2$  then put  $(h_k, f_k)$  into filter & tidy filter up
        set  $x_{k+1} = x_k + d_k$ 
      else
        reduce regularization parameter  $\mu = \mu/2$ 
      end
    else
      put  $(h_k, f_k)$  into filter, tidy filter up & and enter restoration phase to
      find acceptable/compatible point,  $x_{k+1}$ 
    end
  until new  $x_{k+1}$  found
  set  $k = k + 1$ , update gradients  $\nabla f_k, \nabla c_k$  & test for convergence
end

```

Figure 4: FASTr: Filter Active-Set Trust-Region Method

3 Convergence Analysis

The global convergence proof is similar to the one in [25] and shows that the iterates generated by FASTr converge to stationary point. In fact, taking RLP steps is sufficient for global convergence. However, we also show that the convergence of the EQP steps follows from a Cauchy-point type argument similar to [25]. We make the following blanket assumptions:

A1 The functions $f(x), c(x)$ are twice continuously differentiable.

A2 The iterates remain in a compact region $X \neq \emptyset$.

The first assumption is standard. The second assumption can be enforced by adding bounds to all variables. There are four distinct outcomes for the algorithm (see below for a proof that the inner iteration terminates):

OR The restoration phase iterates infinitely and fails to find a point that is acceptable to the filter and for which $\text{RLP}(x_k, \mu)$ is consistent.

OK A KKT point is found.

OF For k sufficiently large, all iterations are of f-type.

OH There exists an infinite number of h-type iterations.

The main convergence theorem will show that for outcomes **OF** and **OH**, the limit point is a Fritz-John (FJ) point. Necessary conditions for x^* to be an FJ point of P are that x^* is feasible and that the set of strictly feasible descent directions is empty. In other words,

$$\{s \mid s^T g^* < 0, \quad s^T a_i^* > 0, \quad i \in \mathcal{A}^*\} = \emptyset, \quad (3.7)$$

where \mathcal{A}^* is the active set at x^* .

If x^∞ is feasible but not an FJ point, then it follows from **A1** and (3.7) that there exist $\epsilon > 0$ and a vector s such that $\|s\| = 1$, and

$$s^T g(x) \leq -\epsilon, \quad \text{and} \quad s^T a_i(x) \geq \epsilon, \quad i \in \mathcal{A}^\infty \quad (3.8)$$

for all x in a neighborhood \mathcal{N}^∞ of x^∞ .

3.1 Preliminary Lemmas

We first derive a useful result following from Taylor's theorem.

Lemma 3.1 *Let Assumptions **A1** and **A2** hold and let $\|g(x)\| \leq M_1$, $\|A(x)\| \leq M_1$, and $s^T \nabla^2 c_i s \leq M_2$ for all $x \in X$. Let $d \neq 0$ solve*

$$(\text{RLP}(x_k, \mu)) \begin{cases} \underset{d}{\text{minimize}} & \mu g_k^T d + \frac{1}{2} d^T d \\ \text{subject to} & c_k + A_k^T d \geq 0, \end{cases}$$

and assume that $\|c(x)\| \leq \mu M_2$ for all x in some neighborhood of x_k . Then it follows that there exists $M > 0$ such that

$$c_i(x_k + d) \geq -\mu^2 M, \quad \forall i = 1, \dots, m \quad (3.9)$$

$$\Delta f \geq \Delta l - \mu^2 M. \quad (3.10)$$

Proof. The proof of this lemma differs slightly from the proof in earlier papers (e.g., [25]) because we do not have an explicit bound on the length of the step. To obtain such a bound, we observe that the Karush-Kuhn-Tucker (KKT) conditions of $\text{RLP}(x_k, \mu)$ imply that

$$d_k = -\mu g_k + A_k y, \quad (3.11)$$

where y are the dual variables. We would like a result that shows that $\|d_k\| = \mathcal{O}(\mu)$. To this end, consider the dual BQP(x_k, μ), and observe that without loss of generality, its solution is given by $y = (y_1, y_2)$, where $y_1 = 0$, and $y_2 \geq 0$ with

$$y_2 = [A_k^T A_k]_{22}^{-1} (c_k - \mu A_k^T g_k).$$

From the assumption that $\|c(x)\| \leq \mu M_2$, it therefore follows that

$$\|y\| \leq M_3 \mu$$

for some $M_3 > 0$ (because $[A_k^T A_k]_{22}^{-1}$ is bounded and $\|\mu A_k^T g_k\| \leq \mu M_1^2$). Substituting this expression into (3.11) shows that there exists $M > 0$ such that

$$\|d_k\| \leq M \mu.$$

We can now proceed with the proof as in earlier papers. Since d_k is feasible, it follows that there exists some z on the line segment between x_k and $x_k + d_k$ such that

$$c_i(x_k + d) = c_i(x_k) + a_i(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 c_i(z) d \geq -M \mu^2,$$

which shows (3.9). Similarly, it follows that

$$\begin{aligned} \Delta f &= f(x_k) - f(x_k + d_k) \\ &= f(x_k) - f(x_k) - g_k^T d_k - \frac{1}{2} d_k^T \nabla^2 f(z) d \\ &\geq \Delta l - \mu^2 M, \end{aligned}$$

which shows (3.10). \square

Note that we use Lemma 3.1 only near feasible points, in which case the assumption $\|c(x)\| \leq \mu M_1$ is always satisfied. Next, we show that the algorithm is well defined and that the inner iteration terminates finitely.

The next lemma shows how solutions of $\text{RLP}(x, \mu)$ change when μ changes but the active set remains unchanged.

Lemma 3.2 *Consider $\text{RLP}(x, \mu)$, and let $d(\mu)$ be its (unique) solution. Let $\mu_2 < \mu_1$, and assume that the active set $\mathcal{A}(\mu) = \mathcal{A}$ is constant for all $\mu_2 \leq \mu \leq \mu_1$. Then it follows that the solutions $d_2 = d(\mu_2)$ and $d_1 = d(\mu_1)$ satisfy*

$$2\mu_2(\mu_1 - \mu_2)s^T s \leq \|d_1\|_2^2 - \|d_2\|_2^2 \leq 2\mu_1(\mu_1 - \mu_2)s^T s, \quad (3.12)$$

where s is the (primal) solution to the auxiliary system

$$\begin{bmatrix} I & A^T \\ -A & 0 \end{bmatrix} \begin{pmatrix} s \\ y \end{pmatrix} = \begin{pmatrix} -g \\ c \end{pmatrix}, \quad (3.13)$$

where A is the Jacobian of the active constraints and c are the residuals of the active constraints.

Proof. Since d_i solves $\text{RLP}(x, \mu_i)$, it follows that

$$\frac{1}{2}d_1^T d_1 + \mu_2 d_1^T g \geq \frac{1}{2}d_2^T d_2 + \mu_2 d_2^T g \quad \text{and} \quad \frac{1}{2}d_2^T d_2 + \mu_1 d_2^T g \geq \frac{1}{2}d_1^T d_1 + \mu_1 d_1^T g, \quad (3.14)$$

which implies that

$$2\mu_2 g^T (d_2 - d_1) \leq \|d_1\|_2^2 - \|d_2\|_2^2 \leq 2\mu_1 (\mu_1 - \mu_2) g^T (d_2 - d_1). \quad (3.15)$$

Let A denote the Jacobian of the active constraints. Then the KKT conditions give that

$$\begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -\mu_1 g \\ c \end{pmatrix} \quad \text{and} \quad \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} -\mu_2 g \\ c \end{pmatrix}.$$

Thus

$$\begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_2 - d_1 \\ y_2 - y_1 \end{pmatrix} = (\mu_2 - \mu_1) \begin{pmatrix} -g \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} d_2 - d_1 \\ y_2 - y_1 \end{pmatrix} = \begin{pmatrix} \bar{s} \\ \bar{y} \end{pmatrix} = (\mu_2 - \mu_1) \begin{pmatrix} s \\ y \end{pmatrix}.$$

Now observe that (3.13) implies that

$$(s, 0)^T \begin{bmatrix} I & -A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} s \\ y \end{pmatrix} = -g^T s = s^T s,$$

which implies that $g^T (d_2 - d_1) = (\mu_1 - \mu_2) s^T s$, which combines with (3.15) to give the result. \square

This lemma suggests an alternative to (2.6) to trigger the restoration phase. Let $\mu_1 = \mu$ and $\mu_2 = \mu/2$. Then it follows that

$$1 - \mu^2 \frac{s^T s}{\|d(\mu)\|_2^2} \leq \frac{\|d(\mu/2)\|_2^2}{\|d(\mu)\|_2^2} \leq 1 - \frac{\mu^2}{2} \frac{s^T s}{\|d(\mu)\|_2^2} \quad (3.16)$$

provided that the active set remains unchanged (a fact that can be checked cheaply). Thus, we could enter restoration whenever two consecutive RLP steps d_1 and d_2 for $\mu_1 > \mu_2$ have similar length. For example, if

$$\frac{\|d_2\|}{\|d_1\|} \geq 0.99.$$

Another important question concerns the relationship between $\text{RLP}(x, \mu)$ and the classical trust-region subproblem

$$(\text{TR}(x, \rho)) \begin{cases} \text{minimize} & g^T d \\ \text{subject to} & c + A^T d \geq 0, \\ & \|d\|_2 \leq \rho, \end{cases}$$

where $\rho > 0$ is the standard trust-region radius. We would not want to solve $\text{TR}(x, \rho)$ in practice (*SLIQUE* uses an ℓ_∞ norm). Some interesting connections exist between the two TR subproblems.

Corollary 3.3 *The solution to $RLP(x, \mu)$ is a continuous piecewise linear path as a function of μ .*

Proof. There exist a finite number of breakpoints along which the active set of $RLP(x, \mu)$ is unchanged. Consider the active set for two values of μ for which the active set is the same. Then it follows that the active set is the same for all values of μ in between. \square

Lemma 3.4 *Let \mathcal{A} be the active general constraints at the solution $RLP(x, \mu)$, and consider the following two equality constraint problems for a fixed active set \mathcal{A} :*

$$(P_P) \begin{cases} \underset{d}{\text{minimize}} & \mu g^T d + \frac{1}{2} d^T d \\ \text{subject to} & A_{\mathcal{A}}^T d = -c_{\mathcal{A}} \end{cases} \quad \text{and} \quad (P_T) \begin{cases} \underset{d}{\text{minimize}} & g^T d \\ \text{subject to} & A_{\mathcal{A}}^T d = -c_{\mathcal{A}} \\ & \frac{1}{2} \|d\|_2^2 = \frac{\rho^2}{2}. \end{cases}$$

Then it follows that

$$\rho^2 = \|d_c\|_2^2 + \mu^2 \|d_g\|_2^2, \quad (3.17)$$

where d_c is the minimum norm solution to $A_{\mathcal{A}}^T d = -c_{\mathcal{A}}$ and d_g is the projection of $-g$ onto the null-space of $A_{\mathcal{A}}^T$.

Proof. Let $A = A_{\mathcal{A}}$ for simplicity. The solution of P_P satisfies

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_p \\ z_p \end{pmatrix} = \begin{pmatrix} -\mu g \\ -c \end{pmatrix}.$$

Define (d_c, z_c) as the solution of the system,

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_c \\ z_c \end{pmatrix} = \begin{pmatrix} 0 \\ -c \end{pmatrix}.$$

That is, d_c is the minimum norm solution to $A_{\mathcal{A}}^T d = -c_{\mathcal{A}}$. Also, define (d_g, z_g) as the solution of the system,

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_g \\ z_g \end{pmatrix} = \begin{pmatrix} -g \\ 0 \end{pmatrix}.$$

That is, d_g is the projection of $-g$ onto the null-space of $A_{\mathcal{A}}^T$. Then it follows that

$$\begin{pmatrix} d_p \\ z_p \end{pmatrix} = \begin{pmatrix} d_c \\ z_c \end{pmatrix} + \mu \begin{pmatrix} d_g \\ z_g \end{pmatrix}.$$

Therefore,

$$\|d_p\|_2^2 = \|d_c\|_2^2 + 2\mu d_c^T d_g + \mu^2 \|d_g\|_2^2. \quad (3.18)$$

Using the definition of d_c , we observe that

$$0 = (d_g^T : 0^T)^T \begin{pmatrix} 0 \\ -c \end{pmatrix} = (d_g^T : 0^T)^T \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} d_c \\ z_c \end{pmatrix} = (d_g^T : d_g^T A)^T \begin{pmatrix} d_c \\ z_c \end{pmatrix} = d_g^T d_c,$$

where the final equality follows from the fact that $A^T d_g = 0$. Substituting this into (3.18), we get that

$$\|d_p\|_2^2 = \|d_c\|_2^2 + \mu^2 \|d_g\|_2^2.$$

Thus, the problems P_P and P_T are equivalent if $\|d_p\|_2^2 = \rho^2$, which proves the result. \square

Lemma 3.5 *Let Assumptions A1 and A2 hold. Then the inner iteration terminates finitely.*

Proof: Clearly, if the inner iteration does not terminate finitely, then the rule for decreasing μ will ensure that $\mu \rightarrow 0$. We distinguish two cases depending on whether $h_k > 0$ or $h_k = 0$.

If $h_k > 0$, we consider the limiting ($\mu \rightarrow 0$) QP

$$\begin{cases} \text{minimize}_d & \frac{1}{2}d^T d \\ \text{subject to} & c_k + A_k^T d \geq 0 \end{cases}$$

and let its solution be d_0 . Since $h_k > 0$, it follows that $d_0 \neq 0$. As $\mu \rightarrow 0$, it follows that the solution $d(\mu)$ of $\text{RLP}(\mu, x_k)$ approaches d_0 . In fact, the solution changes linearly, as μ changes, and there exist a finite number of breakpoints by Lemma 3.2 and Corollary 3.3. We can readily detect whether reducing μ results in changes to the active set, and we use this mechanism as a trigger for our restoration phase.

If $h_k = 0$, then we have shown that solving $\text{RLP}(\mu, x_k)$ is equivalent to solving $\text{TR}(\rho, x_k)$ for sufficiently small μ and a suitable $\rho > 0$; see Lemma 3.4. Thus, finiteness of the inner iteration follows now from the result in [25]. \square

The next lemma shows that if infinitely many points are added to the filter, then the limit must be feasible. This result is proved in [6].

Lemma 3.6 *Consider an infinite sequence of iterations on which (h_k, f_k) is entered into the filter and f_k is bounded below. It follows that $h_k \rightarrow 0$.*

Proof. See [6, Lemma 1]. \square

Lemma 3.7 *The predicted reduction $\Delta l_k = -g_k^T d$ of $\text{RLP}(x_k, \mu)$ increases monotonically as μ increases.*

Proof. Consider $\text{RLP}(x, \mu_i)$ for two values $\mu_1 < \mu_2$, and let d_i denote their respective solutions. It follows from (3.14) that

$$\mu_1 g_k^T d_1 \leq \mu_1 g_k^T d_2 + \mu_2 (g_k^T d_1 - g_k^T d_2),$$

which in turn implies that

$$(\mu_1 - \mu_2) (g_k^T d_1 - g_k^T d_2) \leq 0.$$

Because $\mu_1 < \mu_2$, it follows that $g_k^T d_1 \geq g_k^T d_2$. Therefore, $\Delta l_k = -g_k^T d$ increases monotonically as μ increases. \square

The next lemma summarizes some useful results about the Cauchy step.

Lemma 3.8 *Let standard assumption hold, and let Δl be the reduction predicted by $\text{RLP}(x_k, \mu)$ (with solution d_{LP}). Then it follows that the Cauchy step satisfies*

$$\Delta q_C \geq \alpha_C \Delta l. \tag{3.19}$$

Moreover, if $\Delta l \geq \mu \epsilon$ for some $\epsilon > 0$, then it follows that

$$\alpha_C \geq \frac{\mu \epsilon}{M}, \tag{3.20}$$

where $M \geq d^T H d, \forall d$.

Proof. The first part, (3.19), follows from [6, Lemma 3]. To show (3.20), set $b := d_{LP}^T H_k d_{LP}$. If $b \leq 0$, then we set $\alpha_C = 1$, and the result follows trivially. Otherwise, consider

$$q(\alpha) = f_k + \alpha g_k^T d_{LP} + \frac{1}{2} \alpha^2 d_{LP}^T H d_{LP},$$

and form the first-order condition with respect to α , which gives

$$\alpha_C = \frac{-g_k^T d}{d_{LP}^T H d_{LP}} \geq \frac{-g_k^T d}{M} = \frac{\Delta l}{M} \geq \frac{\mu \epsilon}{M},$$

where the first inequality follows from $M \geq d^T H d$. \square

3.2 Main Convergence Result

A consequence of Lemma 3.5 is that if the algorithm does not terminate with **OR** or **OK**, then there exists an infinite sequence of type **OF** or **OH**. The following theorem completes the proof by showing that the iterates have an accumulation point that is an FJ point. The proof uses the following two key ideas. First, we show that the limit is feasible. Then, we show that if the limit is bounded away from an FJ point, then the conditions for an f-type step must eventually hold for some μ . This property is used in two ways. In case **OF**, we use a standard argument from unconstrained optimization to show that if the limit is not an FJ point, then f is unbounded below. In case **OH**, we use the fact to show that there cannot be a subsequence of h-type steps converging to a non-FJ point.

Theorem 3.9 *If Assumptions A1 and A2 hold, then for our Algorithm either **OR** or **OK** occurs or the iterates have an accumulation point that satisfies the FJ conditions (3.7).*

Proof. We consider only the case where **OR** or **OK** does not occur. Since the inner loop is finite (Lemma 3.5), the outer iteration generates an infinite sequence, $k \in \mathcal{S}$. Since all iterates lie in the compact set X , there exist an accumulation point x^∞ , and we can assume that $x_k \rightarrow x^\infty$ for $k \in \mathcal{S}$ (after possibly taking a subsequence). The proof that x^∞ is an FJ point is in two parts. First, we show that x^∞ is feasible and then that it is also stationary.

In case **OF**, the sequence \mathcal{S} is just the tail of the main sequence after the last h-type iteration. Note that f_k is monotonically decreasing and bounded below. Hence, it follows that $\sum_{k \in \mathcal{S}} \Delta f_k$ is convergent. Because the iterations are f-type iterations, it follows that

$$\Delta f_k \geq \sigma \Delta l_k \geq \sigma \delta (h_k)^2.$$

The convergence of $\sum_{k \in \mathcal{S}} \Delta f_k$ implies that $h_k \rightarrow 0$. Thus, in case **OF**, any accumulation point of the algorithm is feasible.

In case **OH**, we start by considering the thinner subsequence of h-type iterations, on which (h_k, f_k) is entered into the filter. Lemma 3.6 shows that $h_k \rightarrow 0$.¹

¹Note that we cannot get feasibility for every limit point (or FJ conditions for every limit point) because we are not adding every point to the filter. For example, we may have a sequence of alternating f/h-type iterations. Lemma 3.6 shows that the sequence of h-type iterations converges to zero, but the f-type iterations need not be monotonic. Hence, the argument from the first part would fail.

Now we assume that x^∞ is not an FJ point and seek a contradiction. Since x^∞ is not an FJ point, it follows that there exists a direction s for all x in some neighborhood $\mathcal{N}^\infty := \mathcal{N}(x^\infty)$ of x^∞ such that (3.8) holds. For k sufficiently large, we consider the effect of a step μs in $\text{RLP}(x_k, \mu)$.

The remainder of the proof is similar to the proof in [6]. For active constraints at x^∞ we have from (3.8) that

$$c_i^{(k)} + \mu a_i^{(k)T} s \geq -h_k + \mu\epsilon \quad i \in \mathcal{A}^\infty. \quad (3.21)$$

For inactive constraints $i \notin \mathcal{A}^\infty$, if $k \geq K$ is sufficiently large, then there exist positive constants \bar{c} and \bar{a} , independent of k , such that

$$c_i^{(k)} \geq \bar{c} \quad \text{and} \quad a_i^{(k)T} s \geq -\bar{a},$$

by continuity of c_i and boundedness of a_i on X . It follows that

$$c_i^{(k)} + \mu a_i^{(k)T} s \geq \bar{c} - \mu\bar{a} \quad i \notin \mathcal{A}^\infty. \quad (3.22)$$

If we denote $\kappa = \bar{c}/\bar{a} > 0$, it follows for $k \geq K$ that if

$$h_k/\epsilon \leq \mu \leq \kappa, \quad (3.23)$$

then from (3.21) and (3.22),

$$c_i^{(k)} + \mu a_i^{(k)T} s \geq 0 \quad i = 1, 2, \dots, m.$$

Thus, if (3.23) holds, we are assured that μs is a feasible step and hence that $\text{RLP}(x_k, \mu)$ is a feasible subproblem. It also follows by optimality of d that

$$\Delta l = -g_k^T d \geq -\mu g_k^T s \geq \mu\epsilon \quad (3.24)$$

from (3.8).

Thus, if $\mu^2 \leq \beta\tau_k/M$, then it follows from (3.9) that $h(c(x_k + d)) \leq \beta\tau_k$. Also we deduce from (3.10) and (3.24) that

$$\frac{\Delta f}{\Delta l} \geq 1 - \frac{\mu^2 M}{\Delta l} \geq 1 - \frac{\mu M}{\epsilon},$$

so if $\mu \leq (1 - \sigma)\epsilon/M$, it follows that $\Delta f \geq \sigma\Delta l$. Combining these results with (3.23), we see for sufficiently large k that if μ satisfies

$$\frac{h_k}{\epsilon} \leq \mu \leq \min \left\{ \frac{(1 - \sigma)\epsilon}{M}, \sqrt{\frac{\beta\tau_k}{M}}, \kappa \right\}, \quad (3.25)$$

then $LP(x_k, \mu)$ is compatible, $h(c(x_k + d)) \leq \beta\tau_k$, and $\Delta f \geq \sigma\Delta l$. Also from (3.24) and (3.25), $\Delta l \geq h_k$, and hence $\Delta l \geq \delta(h_k)^2$. Thus, for sufficiently large $k \in \mathcal{S}$ there is a range of values (3.25) that guarantee that $x_k + d$ is acceptable to the filter, and the conditions for an f-type step are satisfied.

If the subsequence \mathcal{S} arises from case **OF**, then τ_k is fixed, and the right-hand side of (3.25) is just a number, $\bar{\mu}$ say, while the left-hand side converges to zero. Thus, for sufficiently

large k we can guarantee that a value $\mu_k > \frac{1}{2}\bar{\mu}$ will be chosen in the inner iteration. If we accept the RLP step, then we deduce from (3.24) that $\Delta f_k > \frac{1}{2}\sigma\epsilon\bar{\mu}$. If we take the EQP step, then it follows from Lemma 3.8 that $\Delta f_k \geq \sigma\Delta q_C \geq \sigma\frac{\bar{\mu}^2\epsilon^2}{M}$. In both cases, because an f-type step is taken, we obtain a contradiction to the fact that $\sum_{k \in \mathcal{S}} \Delta f_k$ is convergent.

Finally we look at case **OH**. Because $h_k \rightarrow 0$, it follows that $\tau_k \rightarrow 0$, and there is an infinite subsequence of \mathcal{S} for which $\tau_{k+1} = h_k < \tau_k$. On this subsequence, for sufficiently large k , the range (3.25) becomes

$$\frac{h_k}{\epsilon} \leq \mu \leq \sqrt{\frac{\beta\tau_k}{M}}. \quad (3.26)$$

In the limit, because $h_k < \tau_k$, the upper bound in (3.26) is more than twice the lower bound. Hence, reducing μ in the inner loop will eventually locate a value in the range (3.26) or to the right of that interval. This implies that an f-type step will be taken. If we accept an EQP step, then it follows that this must also be an f-type step, because $\Delta l \geq \delta(h_k)^2$. From Lemma 3.7 it follows that it is not possible for a larger value of μ to produce an h-type step. Thus there exists a k sufficiently large for which an f-type step will be taken. This result contradicts the fact that case **OF** is formed by a subsequence of h-type steps.

Thus a contradiction has been obtained in both case **OF** and case **OH**, and the theorem is proved. \square

4 Properties of Trust-Region Subproblem

This section summarizes some properties of the regularized LP subproblem $\text{RLP}(x, \mu)$.

4.1 Active Set Identification Properties of RLP

This section shows that the RLP identifies the correct active set in a neighborhood of a nondegenerate KKT point for a range of regularization parameters μ .

Proposition 4.1 *Let x_* be a solution to the NLP P at which LICQ and strict complementarity holds, let $\mathcal{A}_* := \{i : c_i(x_*) = 0\}$ be the corresponding active set, let $\mathcal{I}_* := \{1 \dots, m\} / \mathcal{A}_*$ be its complement, and partition the constraints into active and inactive constraints according to \mathcal{A}_* denoted by $c(x) = (c_A(x), c_I(x))^T$.*

Then it follows that for any sufficiently small $\epsilon > 0$ (maximum distance to any active constraint) and suitable $\tau > 0$ (distance to nearest inactive constraint) there exists a neighborhood

$$\mathcal{N}_{\epsilon, \tau}(x_*) := \{x | c_I(x) \geq \tau\epsilon \text{ and } \|c_A(x)\| \leq \epsilon \text{ and } \|x_* - x\| \leq \epsilon\} \quad (4.27)$$

such that for every $x \in \mathcal{N}(x_)$ the regularized LP $\text{RLP}(x, \mu)$ identifies \mathcal{A}_* for all*

$$\mu \in \left[\frac{\kappa_c \epsilon}{\kappa_g}, \frac{\tau - \kappa_f \epsilon}{\kappa_a} \right], \quad (4.28)$$

where the constants $\kappa_c, \kappa_g, \kappa_a > 0$ are independent of τ and ϵ . Moreover, this range is nonempty, as the lower bound converges to zero while the upper bound is a constant.

Proof. We consider the KKT condition for RLP for the active set \mathcal{A}_* and show that this choice of active set is indeed optimal. Thus, consider

$$\begin{bmatrix} I & -A_A \\ A_A^T & 0 \end{bmatrix} \begin{pmatrix} d \\ y_A \end{pmatrix} = \begin{pmatrix} -\mu g \\ -c_A \end{pmatrix},$$

where we have used the same partition into active/inactive constraints. LICQ and the fact that $x \in \mathcal{N}(x_*)$ imply that the KKT matrix is nonsingular, which implies that

$$\begin{pmatrix} d \\ y_A \end{pmatrix} = \begin{bmatrix} I & -A_A \\ A_A^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} -\mu g \\ 0 \end{pmatrix} + \begin{bmatrix} I & -A_A \\ A_A^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ -c_A \end{pmatrix} =: \mu \begin{pmatrix} d_g \\ y_g \end{pmatrix} + \begin{pmatrix} d_c \\ y_c \end{pmatrix}.$$

Now, we derive conditions on μ that ensure that this solution is indeed optimal for RLP. We start by considering the feasibility of d in RLP. Clearly, d satisfies the constraints \mathcal{A}_* . Now consider $\mathcal{I}_* := \{1, \dots, m\} / \mathcal{A}_*$, and observe that d is feasible in RLP if

$$A_I^T (\mu d_g + d_c) + c_I \geq 0 \Rightarrow \mu \leq \min_{i \in \mathcal{I}_* : a_i^T d_g < 0} \frac{c_i + a_i^T d_c}{-a_i^T d_g}, \quad (4.29)$$

where we use the convention that the upper bound on μ is infinite if the condition on the right-hand side of (4.29) is empty.

Similarly, we obtain a bound on μ , by considering dual feasibility:

$$\mu y_g + y_c \geq 0 \Rightarrow \mu \geq \max \left(0, \max_{i \in \mathcal{A}_*} \frac{-y_{c_i}}{y_{g_i}} \right). \quad (4.30)$$

It remains to be shown that these bounds are compatible. To this end, we derive a lower bound on (4.29) and an upper bound on (4.30).

The fact that $\|c_A(x)\| \leq \epsilon$ implies that $-y_{c_i} \leq \kappa_c \epsilon$, and strict complementarity implies that there exists $\kappa_g > 0$ such that $y_{g_i} \geq \kappa_g$. Combining these two observations, we have that (4.30) holds for any

$$\mu \geq \frac{\kappa_c \epsilon}{\kappa_g}. \quad (4.31)$$

Next, we observe that $\|c_A\| \leq \epsilon$, the boundedness of $\|a_i\|$ and g , and the boundedness of the inverse of the augmented system imply

$$a_i^T d_c \geq -\|a_i\| \cdot \|d_c\| \geq -\kappa_f \epsilon.$$

This, together with $c_i \geq \tau \forall i \in \mathcal{I}_*$, and $-a_i^T d_g \leq \kappa_a$ (which follow from the boundedness of g , and LICQ), implies that (4.29) holds for any

$$\mu \leq \frac{\tau - \kappa_f \epsilon}{\kappa_a}. \quad (4.32)$$

We note that we can make ϵ sufficiently small, while τ is a constant that is bounded away from zero. Hence, it follows that the bound in (4.32) is bounded away from zero, while the lower bound, (4.31) can be made arbitrarily small. Combining (4.31) and (4.32), we have that RLP(x, μ) identifies the correct active set \mathcal{A}_* for all μ in (4.28). \square

We remark that the lower bound on μ arises from dual feasibility, that is, from requiring a minimum contribution of the gradient g to identify the correct active set, while the upper bound arises from the inactive constraints.

4.2 Relationship between TR Parameters ρ and μ

Consider the two TR subproblems $RLP(x, \mu)$ and $TR(x, \rho)$ given above. Let $y \geq 0$ be the multipliers of the general constraints $c + A^T d \leq 0$, and let $z \geq 0$ be the multiplier of the trust-region $\|d\|_2 \leq \rho$. Then the KKT conditions of $RLP(x, \mu)$ and $TR(x, \rho)$ are given by

$$\begin{array}{r|l} (RLP(x, \mu)) & (TR(x, \rho)) \\ \hline \mu g + d + Ay = 0 & g + d + Ay = 0 \\ 0 \leq y \perp c + A^T d \leq 0 & 0 \leq y \perp c + A^T d \leq 0 \\ & 0 < z \perp \|d\|_2 = \rho, \end{array}$$

where we have assumed that $TR(x, \rho)$ is consistent and that the TR is strongly active (this will hold for a range of ρ). We can now easily show the following result.

Lemma 4.2 *Assume that $TR(x, \rho)$ is consistent, that $\|d\|_2 = \rho$ is strongly active, and that $\mu > 0$. Then the following conditions exist.*

1. *If (d, y) solves $RLP(x, \mu)$, then $(d, y/\mu)$ solves $TR(x, \rho)$ with $\rho = \|d\|$, and $z = \mu^{-1}$.*
2. *If (d, y) solves $TR(x, \rho)$, then $(d, y\mu)$ solves $RLP(x, \mu)$ with $\mu = z^{-1}$.*

Proof. Compare the KKT conditions. □

5 Conclusions

We have presented a new active-set method for NLPs. Our algorithm identifies the active set by solving a regularized LP, and then solves an equality constrained QP to speed convergence. Global convergence is promoted through the use of a filter.

We have shown that the algorithm converges globally from remote starting points under reasonable assumptions and that the optimal active set is identified for a finite value of the regularization parameter within a neighborhood of a regular point. We have also shown that the regularized LP is equivalent to an ℓ_2 trust-region LP.

The new method avoids some of the pitfalls of other trust-region SQP and SLQP methods. The regularization makes it less likely that an LP becomes infeasible and that feasibility restoration must be invoked. The regularization also avoids a computational disadvantage of SLP methods that often require many unnecessary pivots to sort out the optimal trust-region bounds.

One advantage of the new method is the fact that the main computational tasks can be implemented by using iterative solvers. Moreover, there exist efficient parallel implementations of these solvers (e.g., TAO and PETSc) that allow us to develop parallel solver for NLPs based on the computational kernels. We believe that this is an important and valuable property.

Our method triggers feasibility restoration either if $RLP(x_k, \mu)$ is inconsistent or if active set does not change as $\mu \rightarrow 0$. We can detect this easily by taking the solution to BQP or RLP for a given $\bar{\mu}$, and re-solving for the same active set with $\mu = 0$. If this solution is also feasible and optimal, then we know that reducing μ will not give a better point, and

we enter restoration. Otherwise, there exist other active sets that we can explore as $\mu \rightarrow 0$ from $\bar{\mu}$. Note that we can even find the breakpoints (see (2.6)).

Some open questions remain, not least a successful implementation. The most important open question is how to define a good preconditioner for solving $\text{BQP}(x, \mu)$. Partial factorizations such as ILU may provide good preconditioners. However, we believe that the choice of a good preconditioner is problem-dependent and would be difficult to answer in the general context of the present paper. Another important question is how the ordering affects our iterative solvers.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract No. DE-AC02-06CH11357.

References

- [1] M. Anitescu. A superlinearly convergent sequential quadratically constrained quadratic programming algorithm for degenerate nonlinear programming. *SIAM Journal on Optimization*, 12(4):949–978, 2002.
- [2] H. Y. Benson and D. Shanno. Interior-point methods for nonconvex nonlinear programming: Regularization and warmstarts. *Computational Optimization and Applications*, page to appear, 2007.
- [3] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming, Series B*, 100(1):27–48, 2004.
- [4] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. On the convergence of successive linear-quadratic programming algorithms. Technical Report RAL-TR-2004-032, Rutherford Appleton Laboratory, 2004.
- [5] R. H. Byrd, J. Nocedal, and R. Waltz. KNITRO: An integrated package for nonlinear optimization. Technical report, Department of Electrical Engineering and Computer Science, Northwestern University, 2005.
- [6] C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177, 2003.
- [7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25(2):433–460, 1988. See also same journal 26:764–767, 1989.
- [8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*. Springer Verlag, Heidelberg, 1992.

- [9] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.
- [10] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270, 2002.
- [11] J. Fliege, C. Heermann, and D. Molz. An adaptive primal-dual warm-start technique for quadratic multiobjective optimization. Technical Report 2006/34, School of Mathematics, University of Birmingham, 2006.
- [12] M. Fukushima, Z.-Q. Luo, and P. Tseng. A sequential quadratically constrained quadratic programming method for differentiable convex minimization. *SIAM Journal on Optimization*, 13(4):1098–1119, 2003.
- [13] J. Gondzio and A. Grothey. A new unblocking technique to warmstart interior point methods based on sensitivity analysis. Technical Report MS-06-005, School of Mathematics, University of Edinburgh, 2006.
- [14] N. I. M. Gould, D. Orban, and Ph. L. Toint. Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, 14:299–361, 2005.
- [15] N. I. M. Gould and Ph. L. Toint. A quadratic programming bibliography. Numerical Analysis Group Internal Report 2000-1, Rutherford Appleton Laboratory, 2000. See also <ftp://ftp.numerical.rl.ac.uk/pub/qpbook/qpbook.bib>.
- [16] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal of Optimization*, 2(1):121–152, 1992.
- [17] S. Kruk. Semidefinite programming applied to nonlinear programming. Master’s thesis, University of Waterloo, Ontario, Canada, 1996.
- [18] S. Kruk and H. Wolkowicz. SQ²P, sequential quadratic constrained quadratic programming. In Y. Yuan, editor, *Advances in Nonlinear Programming*, pages 177–204, Dordrecht, The Netherlands, 1998. Kluwer Academic Publishers.
- [19] C. Lin and J. J. Moré. Newton’s method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.
- [20] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.
- [21] O. L. Mangasarian. Iterative solution of linear programs. *SIAM Journal on Numerical Analysis*, 18(4):606–614, 1981.
- [22] O. L. Mangasarian and R. R. Meyer. Nonlinear perturbations of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.
- [23] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

- [24] R. Fletcher and E. Sainz de la Maza. Nonlinear Programming and nonsmooth Optimization by successive Linear Programming. *Mathematical Programming*, 43:235–256, 1989.
- [25] R. Fletcher, S. Leyffer and Ph. L. Toint. On the global convergence of an SLP-filter algorithm. Numerical Analysis Report NA/183, University of Dundee, August 1998.
- [26] M. V. Solodov. On the sequential quadratically constrained quadratic programming methods. *Mathematics of Operations Research*, 29(1):64–79, 2004.
- [27] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. Halstead Press, Wiley, New York, 1977.
- [28] H. Yamashita and H. Dan. Global convergence of a trust region sequential quadratic programming method. *Journal of the Operations Research Society of Japan*, 48(1):41–56, 2005.
- [29] Y.-B. Zhao and D. Li. Locating the least 2-norm solution of linear programs via a path-following methods. *SIAM J. Optimization*, 12(4):893–912, 2002.