ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Foundations of Constrained Optimization

Sven Leyffer and Ashutosh Mahajan

Mathematics and Computer Science Division

Preprint ANL/MCS-Pnnnn-mmmm

May 27, 2010

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Contents

Foundations of Constrained Optimization*

Sven Leyffer† and Ashutosh Mahajan ‡

May 27, 2010

Abstract

This chapter presents the foundations of nonlinearly constrained optimization, or nonlinear programming. We emphasize general methods and highlight their key components, such as local model and global convergence mechanism. We summarize convergence results.

Keywords: Mathematical programming methods, Newton-type methods, nonlinear programming, interior-point methods, sequential quadratic programming, sequential linear programming

AMS-MSC2000: 49M05, 49M15, 49M37, 65K05, 90C30, 90C51, 90C55

1 Background and Introduction

Nonlinearly constrained optimization problems (NCOs) are an important class of problems with a broad range of engineering, scientific, and operational applications. For ease of presentation, we consider NCOs of the form

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) = 0 \;\; \text{and} \;\; x \geq 0, \tag{1.1}$$

where the objective function, $f : \mathbb{R}^n \to \mathbb{R}$, and the constraint functions, $c : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable. We denote the multipliers corresponding to the equality constraints, $c(x) = 0$, by y and the multipliers of the inequality constraints, $x \geq 0$, by $z \geq 0$. An NCO may also have unbounded variables, upper bounds, or general range constraints of the form $l_i \leq c_i(x) \leq u_i$, which we omit for the sake of simplicity.

In general, one can not solve (1.1) directly or explicitly. Instead, an iterative method is used that solves a sequence of simpler, approximate subproblems to generate a sequence of approximate solutions, $\{x_k\}$, starting from an initial guess x_0. A simplified algorithmic framework for solving (1.1) is as follows.

*Preprint ANL/MCS-Pnnnn-mmmm.

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, leyffer@mcs.anl.gov.

‡Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, mahajan@mcs.anl.gov.

Given initial estimate $x_0 \in \mathbb{R}^n$, set $k = 0$;
while x_k *is not optimal* **do**
  repeat
    Approximately solve and refine a local model of (1.1) around x_k.
  until *an improved solution estimate* x_{k+1} *is found* ;
  Check whether x_{k+1} is optimal; set $k = k + 1$.
end

Algorithm 1: Framework for Nonlinear Optimization Methods

In this chapter, we review the basic components of methods for solving NCOs. In particular, we review the four fundamental components of Algorithm 1: *convergence test* that checks for optimal solutions, or detects failure; *local model* that computes an improved new iterate; *globalization strategy* that ensures convergence from remote starting points, by indicating whether a new solution estimate is better than the current estimate; and *globalization mechanism* that truncates the local model to enforce the globalization strategy, effectively refining the local model.

Algorithms for NCOs are categorized by the choice they implement for each of these fundamental components. In the next section, we review the fundamental building blocks of methods for nonlinearly constrained optimization.

Notation: Throughout this section, we denote iterates by $x_k, k = 1, 2, \ldots$, and we use subscripts to indicate functions evaluated at an iterate, e.g., $f_k = f(x_k)$ and $c_k = c(x_k)$. We also denote the gradients by $g_k = \nabla f(x_k)$, and $A_k = \nabla c(x_k)$. The Hessian of the Lagrangian is denoted by H_k.

2 Convergence Test

We start by describing the convergence test, a common component among all NCO algorithms. The convergence test also provides the motivation for many local models that are described next. The convergence analysis of NCO algorithms typically provides convergence only to KKT points. A suitable approximate convergence test is thus given by

$$\|c(x_k)\| \leq \epsilon \text{ and } \|g_k - A_k y_k - z_k\| \leq \epsilon \text{ and } \|\min(x_k, z_k)\| \leq \epsilon,$$

where $\epsilon > 0$ is the tolerance and the $\min$ in the last expression corresponding to complementary slackness is taken componentwise.

In practice, it may not be possible to ensure convergence to an approximate KKT point, for example, if the constraints fail to satisfy a constraint qualification (Mangasarian, 1969, Ch. 7). In that case, we replace the second condition by

$$\|A_k y_k + z_k\| \leq \epsilon,$$

which corresponds to a Fritz-John point.

Infeasible Stationary Points Unless the NCO is convex or some restrictive assumptions are made, methods cannot guarantee convergence even to a feasible point. Moreover, an NCO may not even have a feasible point, and we are interested in a (local) certificate of infeasibility. In this case, neither the local model nor the convergence test is adequate to achieve and detect convergence. A more appropriate convergence test and local model can be based on the following feasibility problem:

$$\underset{x}{\text{minimize}} \ \|c(x)\| \quad \text{subject to} \ \ x \geq 0, \tag{2.2}$$

which can be formulated as a smooth optimization problem by introducing slack variables. Local models can be derived analogous to those of Section 3. In general, we can replace this objective by any weighted norm. A suitable convergence test is then

$$\|A_k y_k - z_k\| \leq \epsilon \ \text{ and } \ \|\min(x_k, z_k)\| \leq \epsilon,$$

where y_k are the multipliers or weights corresponding to the norm used in the objective of (2.2). For example, if we use the ℓ_1 norm, then $y_k \in \{-1, 1\}^m$ depending on which side of the equality constraint is active. The multipliers are readily computed as a by-product of solving the local model.

3 Local Model

One key difference among nonlinear optimization methods is how the local model is constructed. The goal of the local model is to provide a step that improves on the current iterate. We distinguish three broad classes of local models: sequential linear models, sequential quadratic models, and interior-point models. Models that are based on the augmented Lagrangian method are more suitably described in the context of merit functions in Section 4.2.

3.1 Sequential Linear and Quadratic Programming

Sequential linear and quadratic programming methods construct a linear or quadratic approximation of (1.1) and solve a sequence of such approximations, converging to a stationary point.

Sequential Quadratic Programming (SQP) Methods: SQP methods successively minimize a quadratic model, $m_k(x)$, subject to a linearization of the constraints about x_k (Han, 1977; Powell, 1978; Boggs and Tolle, 1995). We define the displacement $d := x - x_k$ and obtain the QP

$$\underset{x}{\text{minimize}} \ m_k(d) := g_k^T d + \frac{1}{2} d^T H_k d \quad \text{subject to} \ \ c_k + A_k^T d = 0 \ x_k + d \geq 0, \tag{3.3}$$

where $H_k \simeq \nabla^2 \mathcal{L}(x_k, y_k)$ approximates the Hessian of the Lagrangian and y_k is the multiplier estimate at iteration k. The new iterate is $x_{k+1} = x_k + d$, together with the multipliers y_{k+1} of the linearized constraints of (3.3). The solution of the QP subproblem can become computationally expensive for large-scale problems because the null-space method for solving QPs requires the

factorization of a dense reduced Hessian matrix. If H_k is not positive definite on the null-space of the active constraint normals, then the QP is nonconvex, and SQP methods seek a local minimum of (3.3).

Sequential Linear Programming (SLP) Methods: SLP methods construct a linear approximation to (1.1). In general, this LP will be unbounded, and SLP methods require the addition of a trust-region (discussed in more detail in the next section):

$$\underset{x}{\text{minimize}}\ m_k(d) = g_k^T d \quad \text{subject to } c_k + A_k^T d = 0,\ x_k + d \geq 0,\ \text{ and } \|d\|_\infty \leq \Delta_k, \tag{3.4}$$

where $\Delta_k > 0$ is the trust-region radius. Griffith and Stewart (1961) used this method without using a trust-region but with the assumption that the variables are bounded. In general, $\Delta_k \to 0$ must converge to zero to ensure convergence. SLP methods can be viewed as steepest descent methods and typically converge only linearly. If, however there are exactly n active and linearly independent constraint normals at the solution, then SLP reduces to Newton's method for solving a square system of nonlinear equations and converges superlinearly.

Sequential Linear/Quadratic Programming (SLQP) Methods: SLQP methods combine the advantages of the SLP method (fast solution of the LP) and SQP methods (fast local convergence) by adding an equality-constrained QP to the SLP method (Fletcher and Sainz de la Maza, 1989; Chin and Fletcher, 2003; Byrd et al., 2004). SLQP methods thus solve two subproblems: first, an LP is solved to obtain a step for the next iteration and also an estimate of the active set $\mathcal{A}_k := \left\{ i : [x_k]_i + \hat{d}_i = 0 \right\}$ from a solution $\hat{d}$ of (3.4). This estimate of the active set is then used to construct an equality-constrained QP (EQP), on the active constraints,

$$\underset{x}{\text{minimize}}\, q_k(d) = g_k^T d + \frac{1}{2} d^T H_k d \quad \text{subject to } \ c_k + A_k^T d = 0\ [x_k]_i + d_i = 0,\ \forall i \in \mathcal{A}_k. \tag{3.5}$$

If H_k is second-order sufficient (i.e., positive-definite on the null-space of the constraints), then the solution of (3.5) is equivalent to the following linear system obtained by applying the KKT conditions to the EQP:

$$(EQP) \begin{bmatrix} H_k & -[A_k : I_k] \\ A_k^T & \\ I_k^T & \end{bmatrix} \begin{pmatrix} x \\ y_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} -g_k + H_k x_k \\ -c_k \\ 0 \end{pmatrix},$$

where $I_k = [e_i]_{i \in \mathcal{A}_k}$ are the normals of the active inequality constraints. By taking a suitable basis from the LP simplex solve, SLQP methods can ensure that $[A_k : I_k]$ has full rank. Linear solvers such as MA57 can also detect the inertia; and if H_k is not second-order sufficient, a multiple of the identity can be added to ensure descent.

Sequential Quadratic/Quadratic Programming (SQQP) Methods: SQQP methods have recently been proposed as SQP types of methods. First, a convex QP model constructed by using a positive-definite Hessian approximation is solved, followed by a reduced inequality constrained model or an EQP with the exact second derivative of the Lagrangian.

The methods described in this section are also often referred to as *active-set methods*, because the solution of each LP or QP provides not only a suitable new iterate but also an estimate of the active set at the solution. It can be shown that this estimate settles down under mild assumptions.

Theory of Sequential Linear/Quadratic Programming Methods. If H_k is the exact Hessian of the Lagrangian and if the Jacobian of the active constraints has full rank, then SQP methods converge quadratically near a minimizer that satisfies a constraint qualification and a second-order sufficient condition (Boggs and Tolle, 1995). It can also be shown that, under the additional assumption of strict complementarity, that all four methods identify the optimal active set in a finite number of iterations, and this is one of the motivations for SLQP methods.

3.2 Interior-Point Methods

Interior-point methods (IPMs) are an alternative approach to active-set methods. Interior-point methods are a class of perturbed Newton methods that postpone the decision of which constraints are active until the end of the iterative process. The most successful IPMs are primal-dual IPMs, which can be viewed as Newton's method applied to the perturbed first-order conditions of (1.1):

$$(F_\mu) \qquad 0 = F_\mu(x, y, z) = \begin{pmatrix} \nabla f(x) - \nabla c(x)^T y - z \\ c(x) \\ Xz - \mu e \end{pmatrix},$$

where $\mu > 0$ is the barrier parameter, $X = \text{diag}(x)$ is a diagonal matrix with x along its diagonal, and $e = (1, \ldots, 1)$ is the vector of all ones. Note that, for $\mu = 0$, these conditions are equivalent to the first-order conditions except for the absence of the non-negativity constraints $x, z \geq 0$.

Interior-point methods start at an "interior" iterate $x_0, z_0 > 0$ and generate a sequence of interior iterates $x_k, z_k > 0$ by approximately solving the first-order conditions (F_μ) for a decreasing sequence of barrier parameters. Interior-point methods can be shown to be polynomial-time algorithms for convex NLPs. Newton's method applied to the primal-dual system around x_k gives rise to the local model,

$$\begin{bmatrix} H_k & -A_k & -I \\ A_k^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F_\mu(x_k, y_k, z_k),$$

where H_k approximates the Hessian of the Lagrangian, $\nabla^2 \mathcal{L}_k$, and the step $(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k, y_k, z_k) + (\alpha_x \Delta_x, \alpha_y \Delta_y, \alpha_z \Delta_z)$ is safeguarded to ensure that $x_{k+1}, z_{k+1} > 0$ remain strictly positive.

Relationship to Barrier Methods: Primal-dual interior-point methods are related to earlier barrier methods. These methods were given much attention in the 1960s but soon lost favor because of the ill-conditioning of the Hessian. They regained attention in the 1980s after it was shown

that these methods can work well for linear programming problems. See a survey by Forsgren et al. (2002) for recent developments. Barrier methods approximately solve a sequence of barrier problems,

$$\underset{x}{\text{minimize}} \;\; f(x) - \mu \sum_{i=1}^{n} \log(x_i) \quad \text{subject to} \;\; c(x) = 0, \tag{3.6}$$

for a decreasing sequence of barrier parameters $\mu > 0$. The first-order conditions of (3.6) are given by

$$\nabla f(x) - \mu X^{-1} e - A(x) y = 0 \;\text{ and }\; c(x) = 0. \tag{3.7}$$

Applying Newton's method to this system of equations results in the following local model around x_k:

$$\begin{bmatrix} H_k + \mu X_k^{-2} & -A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} g_k - \mu X_k^{-1} e - A_k y_k \\ c_k \end{pmatrix}.$$

Introducing first-order multiplier estimates $Z(x_k) := \mu X_k^{-1}$, which can be written as $Z(x_k) X_k = \mu e$, we obtain the system

$$\begin{bmatrix} H_k + Z(x_k) X_k^{-1} & -A_k \\ A_k & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} g_k - \mu X_k^{-1} e - A_k y_k \\ c_k \end{pmatrix},$$

which is equivalent to the primal-dual Newton system, where we have eliminated

$$\Delta z = -X^{-1} Z \Delta x - Ze - \mu X^{-1} e.$$

Thus, the main difference between classical barrier methods and the primal-dual IPMs is that Z_k is not free for barrier methods but is chosen as the primal multiplier $Z(x_k) = \mu X_k^{-1}$, which avoids some difficulties with ill-conditioning of the barrier Hessian.

Convergence of Barrier Methods: If there exists a compact set of isolated local minimizers of (1.1) with at least one point in the closure of the strictly feasible set, then it follows that barrier methods converge to a local minimum (Wright, 1992).

4 Globalization Strategy to Promote Convergence from Remote Starting Points

The local improvement models of the preceeding section guarantee convergence only in a small neighborhood of a regular solution. Globalization strategies are concerned with ensuring convergence from remote starting points to stationary points (and should not be confused with global optimization). To ensure convergence from remote starting points, we must monitor the progress of the local method. Monitoring is easily done in unconstrained optimization, where we can measure progress by comparing objective values. In constrained optimization, however we must take the constraint violation into account. Two broad classes of strategies exist: penalty and merit-function, and filter and funnel methods.

4.1 Augmented Lagrangian Methods

The augmented Lagrangian of (1.1) is given by

$$\mathcal{L}(x, y, \rho) = f(x) - y^T c(x) + \frac{\rho}{2}\|c(x)\|_2^2, \tag{4.8}$$

where $\rho > 0$ is the penalty parameter. The augmented Lagrangian is used in two modes to develop algorithms for solving (1.1): by defining a linearly constrained problem or by defining a bound constrained problem.

Linearly constrained Lagrangian methods: These methods successively minimize a shifted augmented Lagrangian subject to a linearization of the constraints. The shifted augmented Lagrangian is defined as

$$\overline{\mathcal{L}}(x, y, \rho) = f(x) - y^T p_k(x) + \frac{\rho}{2}\|p_k(x)\|_2^2, \tag{4.9}$$

where $p_k(x)$ are the higher-order nonlinear terms at the current iterate x_k, that is,

$$p_k(x) = c(x) - c_k - A_k^T(x - x_k). \tag{4.10}$$

This approach results in the following local model:

$$\underset{x}{\text{minimize}}\ \overline{\mathcal{L}}(x, y_k, \rho_k) \quad \text{subject to } c_k + A_k^T(x - x_k) = 0,\ x \geq 0. \tag{4.11}$$

We note that if $c_k + A_k^T(x - x_k) = 0$, then minimizing the shifted augmented Lagrangian is equivalent to minimizing the Lagrangian over these constraints. Linearly constrained, augmented Lagrangian methods solve a sequence of problems (4.11) for a fixed penalty parameter. Multipliers are updated by using a first-order multiplier update rule,

$$y_{k+1} = y_k - \rho_k c(x_{k+1}), \tag{4.12}$$

where x_{k+1} solves (4.11). These methods can be made globally convergent by adding slack variables to deal with infeasible subproblems.

Bound-constrained Lagrangian methods: These methods approximately minimize the augmented Lagrangian,

$$\underset{x}{\text{minimize}}\ \mathcal{L}(x, y_k, \rho_k) \quad \text{subject to } x \geq 0. \tag{4.13}$$

The advantage of this approach is that efficient methods for bound-constrained optimization can readily be applied, such as gradient-projection conjugate-gradient approach, which can be interpreted as an approximate Newton method on the active inequality constraints.

Global convergence is promoted by defining two forcing sequences, $\omega_k \searrow 0$, controlling the accuracy with which every bound constrained problems is solved, and $\eta_k \searrow 0$, controlling progress toward feasibility of the nonlinear constraints. A typical bound constrained Lagrangian method can then be stated as follows:

Given an initial solution estimate (x_0, y_0), and an initial penalty parameter ρ_0.
while x_k *is not optimal* **do**
  Find an ω_k-optimal solution, x_k^c of (4.13). **if** $\|c(x_k^c)\| \leq \eta_k$ **then**
    Perform a first-order multiplier update: $y_{k+1} = y_k - \rho_k c(x_k^c)$
  else
    Increase penalty: $\rho_{k+1} = 10\rho_k$
  end
  Set $k = k + 1$
end

Algorithm 2: Bound Constrained Augmented Lagrangian Method.

Theory of Augmented Lagrangian Methods. Conn et al. (1991) show that a bound-constrained Lagrangian method can globally converge if the sequence $\{x_k\}$ of iterates is bounded and if the Jacobian of the constraints at all limit points of $\{x_k\}$ has column rank no smaller than m. Conn et al. (1991) show that if some additional conditions are met, then their algorithm is R-linearly convergent. Bertsekas (1996) shows that the method converges Q-linearly if $\{\rho_k\}$ is bounded, and superlinearly otherwise.

4.2 Penalty and Merit Function Methods

Penalty and merit functions combine the objective function and a measure of the constraint violation into a single function whose local minimizers correspond to local minimizers of the original problem (1.1). Convergence from remote starting points can then be ensured by forcing descent of the penalty or merit function, using one of the mechanisms of the next section.

Exact penalty functions are an attractive alternative to augmented Lagrangians and are defined as

$$p_\rho(x) = f(x) + \rho\|c(x)\|,$$

where $\rho > 0$ is the penalty parameter. Most approaches use the ℓ_1 norm to define the penalty function. It can be shown that a local minimizer, x^*, of $p_\rho(x)$ is a local minimizer of problem (1.1) if $\rho > \|y^*\|_D$, where y^* are the corresponding Lagrange multipliers, and $\|\cdot\|_D$ is the dual norm of $\|\cdot\|$ (i.e., the ℓ_∞-norm in the case of the ℓ_1 exact-penalty function). Classical approaches using $p_\rho(x)$ have solved a sequence of penalty problems for an increasing sequence of penalty parameters. Modern approaches attempt to steer the penalty parameter by comparing the predicted decrease in the constraint violation to the actual decrease over a step.

Other Merit Functions. A range of other merit functions exist. The oldest, the quadratic penalty function, $f(x)+\rho\|c(x)\|_2^2$, converges only if the penalty parameter diverges to infinity. Augmented Lagrangian functions and Lagrangian penalty functions such as $f(x) + y^T c(x) + \rho\|c(x)\|$ have also

been used to promote global convergence. A key ingredient in any convergence analysis is to connect the local model to the merit function that is being used in a way that ensures a descent property of the merit function; see Section 5.1.

4.3 Filter and Funnel Methods

Filter and funnel methods provide an alternative to penalty methods that does not rely on the use of a penalty parameter. Both methods use step acceptance strategies that are closer to the original problem, by separating the constraints and the objective function.

Filter methods: Filter methods keep a record of constraint violation $h_l := \|c(x_l)\|$ and objective function value $f_l := f(x_l)$ for some previous iterates $x_l, l \in \mathcal{F}_k$ (Fletcher and Leyffer, 2002). A new point is acceptable if it improves either the objective function or the constraint violation compared to all previous iterates. That is, $\hat{x}$ is acceptable, if

$$f(\hat{x}) \leq f_l - \gamma h_l \quad \text{or} \quad h(\hat{x}) \leq \beta h_l \; \forall l \in \mathcal{F}_k,$$

where $\gamma > 0, 0 < \beta < 1$, are constants that ensure iterates cannot accumulate at infeasible limit points. A typical filter is shown in Figure 1 (left), where the straight lines correspond to the region in the (h, f)-plane that is dominated by previous iterations, and the dashed lines correspond to the envelope defined by γ, β.

The filter provides convergence only to a feasible limit because any infinite sequence of iterates must converge to a point, where $h(x) = 0$, provided that $f(x)$ is bounded below. To ensure convergence to a local minimum, filter methods use a standard sufficient reduction condition from unconstrained optimization and employ a switching condition that indicates when this condition should be checked (typically near the feasible set). The sufficient reduction condition is thus

$$f(x_k) - f(x_k + d) \geq -\sigma m_k(d), \tag{4.14}$$

where $\sigma > 0$ is the fraction of predicted decrease and $m_k(d)$ is the model reduction from the local model. This condition is enforced only if the model predicts a decrease in the objective function (far from the feasible set, the local model may indeed predict an increase in the objective, and it would not be appropriate to require an objective decrease). Thus, the switching condition is given by $m_k(d) \geq \gamma h_k^2$. A new iterate that satisfies both conditions is called an f-type iterate, and an iterate for which (4.14) fails is called an h-type iterate to indicate that it mostly reduces the constraint violation. If a new point is accepted, then it is added to the current iterate to the filter, $\mathcal{F}_k$, if $h_k > 0$ or if it corresponds to an h-type iterations (which automatically satisfy $h_k > 0$).

Funnel methods: The method of Gould and Toint (2010) can be viewed as filter methods with just a single filter entry, corresponding to an upper bound on the constraint violation. Thus, the filter contains only a single entry, namely, $(U_k, -\infty)$. The upper bound is reduced during h-type iterations, to force the iterates toward feasibility; and is left unchanged during f-type iterations.

Thus, it is possible to converge without reducing U_k to zero (consistent with the observation that SQP methods converge locally). A schematic interpretation of the funnel is given in Figure 1 (left).

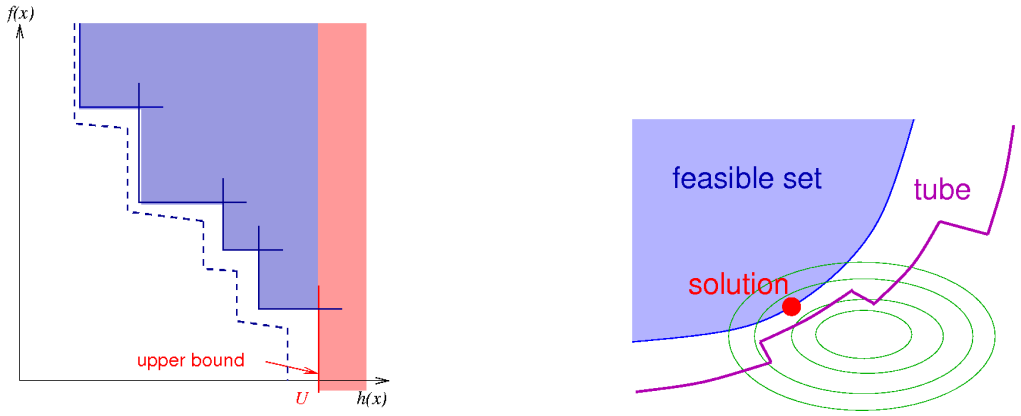

Figure 1: The left figure shows a filter where the shaded area corresponds to the points that are rejected by the filter. The right figure shows a funnel around the feasible set.

4.4 Maratos Effect and Loss of Fast Convergence

One can construct simple examples showing that arbitrarily close to an isolated strict local minimizer, the Newton step will be rejected by the exact penalty function (Maratos, 1978), resulting in slow convergence. This phenomenon is known as the Maratos effect. It can be mitigated by computing a second-order correction step, which is a Newton step that uses the same linear system with an updated right-hand-side (Fletcher, 1987; Nocedal and Wright, 1999). An alternative method to avoid the Maratos effect is the use of nonmonotone techniques that require descent over only the last M iterates, where $M > 1$ is a constant.

5 Globalization Mechanisms

In this section, we review two mechanisms to reduce the step that is computed by the local model: line-search methods and trust-region methods. Both mechanisms can be used in conjunction with any of the local models and any of the global convergence strategies, giving rise to a broad family of algorithms. In Chapter **??**, we describe how these components are used in software for NCOs.

5.1 Line-Search Methods

Line-search methods enforce convergence with a backtracking line-search along the direction $s = \hat{x} - x_k$. It is important to ensure that the model produces a descent direction, e.g., $\nabla\Phi(x_k)^T s < 0$ for a merit or penalty function $\Phi(x)$; otherwise, the line-search may not terminate. A popular line-search is the Armijo search (Nocedal and Wright, 1999) in Algorithm 3 for a merit function $\Phi(x)$.

The algorithm can be shown to converge to a stationary point, detect unboundedness, or converge to a point where there are no directions of descent.

Given initial estimate $x_0 \in \mathbb{R}^n$, let $0 < \sigma < 1$, and set $k = 0$;
while x_k *is not optimal* **do**
  Approximately solve a local model of (1.1) around x_k.
  Denote the solution by $\hat{x}$.
  Let $s := \hat{x} - x_k$
  Make sure that s is a descent direction, e.g. $\nabla\Phi(x_k)^T s < 0$.
  Set $\alpha^0 = 1$ and $l = 0$.
  repeat
    Set $\alpha^{l+1} = \alpha^l/2$ and evaluate $\Phi(x_k + \alpha^{l+1}s)$. Set $l = l + 1$.
  until $\Phi(x_k + \alpha^l s) \leq f_k + \alpha^l \sigma s^T \nabla\Phi_k$;
  set $k = k + 1$.
end

Algorithm 3: (Armijo) Line-Search Method for Nonlinear Optimization

Line-search methods for filters can be defined in a similar way. Instead of checking descent in the merit function, a filter method is used to check acceptance to a filter. Unlike merit functions, filter methods do not have a simple definition of descent; hence, the line-search is terminated unsuccessfully once the step-size α^l becomes smaller than a constant. In this case, filter methods switch to a restoration step.

5.2 Trust-Region Methods

Trust-region methods explicitly restrict the step that is computed by the local model, by adding a trust-region constraint of the form $\|x - x_k\| \leq \Delta_k$ to the local model. Most methods use an ℓ_∞-norm trust-region, which can be represented by bounds on the variables. The trust-region radius, $\Delta_k > 0$, is adjusted at every iteration depending on how well the local model agrees with the NCO, (1.1).

Given initial estimate $x_0 \in \mathbb{R}^n$, choose $\Delta_0 > 0$, and set $k = 0$;
while x_k *is not optimal* **do**
  Reset Δ_k;
  repeat
    Approximately solve a local *trust-region* model with $\|x - x_k\| \leq \Delta_k$.
    Let the solution by $\hat{x}$. **if** $\hat{x}$ *is sufficiently better than* x_k **then**
      Accept the step: $x_{k+1} = \hat{x}$; possibly increase Δ_k.
    else
      Reject the step
      Decrease the trust-region radius, e.g. $\Delta_k = \Delta_k/2$.
    end
  until *an improved solution estimate* x_{k+1} *is found* ;
  Check whether x_{k+1} is optimal; set $k = k + 1$.
end

Algorithm 4: Trust-Region Methods for Nonlinear Optimization

Trust-region methods are related to regularization techniques, which add a multiple of the identity matrix, $\sigma_k I$, to the Hessian, H_k. Locally, the solution of the regularized problem is equivalent to the solution of a trust-region problem with an ℓ_2 trust-region.

Acknowledgments

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. This work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.

References

Bertsekas, D. (1996). *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, Belmont, Mass.

Boggs, P. and Tolle, J. (1995). Sequential quadratic programming. *Acta Numerica*, 4:1–51.

Byrd, R. H., Gould, N. I. M., Nocedal, J., and Waltz, R. A. (2004). An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming, Series B*, 100(1):27–48.

Chin, C. and Fletcher, R. (2003). On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (1991). A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal of Numerical Analysis*, 28(2):545–572.

Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, Chichester.

Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270.

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

Forsgren, A., Gill, P. E., and Wright, M. H. (2002). Interior methods for nonlinear optimization. *SIAM Review*, 4(4):525–597.

Gould, N. I. M. and Toint, P. L. (2010). Nonlinear programming without a penalty function or a filter. *Mathematical Programming*, 122(1):155–196.

Griffith, R. and Stewart, R. (1961). A nonlinear programming technique for the optimization of continuous processing systems. *Management Science*, 7(4):379–392.

Han, S. (1977). A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22(3):297–309.

Mangasarian, O. (1969). *Nonlinear Programming*. McGraw-Hill Book Company, New York.

Maratos, N. (1978). *Exact penalty function algorithms for finite dimensional and control optimization problems*. PhD thesis, University of London.

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer.

Powell, M. (1978). A fast algorithm for nonlinearly constrained optimization calculations. In Watson, G., editor, *Numerical Analysis, 1977*, pages 144–157, Berlin. Springer–Verlag.

Wright, M. (1992). Interior methods for constrained optimization. *Acta Numerica*, 1:341–407.