mization algorithm is required and, in particular, the SQP algorithm must be adapted for this case. Finally, section 9.6 concludes the chapter and provides guides for further reading. Several process examples are also used to illustrate the concepts in this chapter.

9.2 INTRODUCTION TO CONSTRAINED NONLINEAR PROGRAMMING

We consider the nonlinear programming problem, given in general form as:

$$\begin{aligned} \underset{x}{\text{Min}} \quad & f(x) \\ s.t. \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{9.1}$$

where x is an n vector of continuous variables, $f(x)$ is a scalar objective function, $g(x)$ is an m vector of inequality constraint functions, and $h(x)$ is an meq vector of equality constraint functions. These constraints create a region for the variables x, termed the *feasible region,* and we require $n \geq meq$ in order to have any degrees of freedom for optimization. While Eq. (9.1) will be our standard form for nonlinear programs, the NLP problem can be expressed in a number of different ways. For instance, the signs of the objective function and constraint functions could be changed so that we have:

$$\begin{aligned} \underset{x}{\text{Max}} \quad & q(x) \\ s.t. \quad & w(x) \geq 0 \\ & h(x) = 0 \end{aligned} \tag{9.2}$$

for functions defined by $q(x) = -f(x)$ and $w(x) = -g(x)$. Properties of this nonlinear program (NLP) are summarized in Appendix A. In particular, we will develop methods that will find a *local minimum* point x^* for $f(x)$ for a feasible region defined by the constraint functions; that is, $f(x^*) \leq f(x)$ for all x satisfying the constraints in some neighborhood around x^*. Provided that the feasible region is not empty and the objective function is bounded below on this feasible region, we know that such local solutions exist.

On the other hand, finding and verifying *global solutions* to this NLP will not be dealt with in this chapter. In Appendix A, we see that a local solution to the NLP is also a global solution under the following *sufficient* conditions based on convexity. From Appendix A, we define a convex function $\phi(x)$ for x in some domain X, if and only if it satisfies the relation:

$$\phi(\alpha\,\xi + (1-\alpha)\,\eta) \leq \alpha\,\phi(\xi) + (1-\alpha)\,\phi(\eta) \tag{9.3}$$

for any α, $0 \leq \alpha \leq 1$, at all points in ξ and η in X. As derived in Appendix A, sufficient conditions for a global solution for the NLP (9.1) are that:

- the solution is a local minimum for the NLP
- $f(x)$ is *convex*
- $g(x)$ are all *convex*
- $h(x)$ are all *linear*

The last two conditions imply that the feasible region is convex, i.e. for all points ξ and η in the feasible region and for all α, $0 \le \alpha \le 1$, the point $[\alpha\,\xi + (1-\alpha)\,\eta]$ is also in the region. For process optimization, these properties state that any problem with nonlinear equality constraints is nonconvex and in the absence of additional information, there is *no guarantee that a local optimum is global* if these convexity conditions are not met.

To illustrate these concepts, we consider two nonconvex examples that lead to different kinds of solutions.

EXAMPLE 9.1 Optimal Vessel Dimensions

Consider the optimization of a cylindrical vessel with a specified volume. What is the optimal L/D ratio for this vessel that leads to a minimum cost?

The constrained problem can be formulated as one where we minimize a cost based on the amount of material used to make up the top and bottom of the vessel and the sides of the vessel. For a small wall thickness, the amount of material is proportional to the surface area. The cost per area for the materials is given by C_T and C_S for the top and sides, respectively. The specification for volume is written as a constraint and the NLP is given by:

$$\text{Min}\left\{C_T\frac{\pi D^2}{2} + C_S\pi DL = \text{cost}\right\}$$

$$s.t. \quad V - \frac{\pi D^2 L}{4} = 0 \qquad D, L \ge 0 \tag{9.4}$$

Note that for this problem, the feasible region in the variables D and L is nonconvex, because of the nonlinear constraint. We can easily eliminate L from this equation and substitute $L = 4V/\pi D^2$ in the objective function and describe it using the single variable D. Since the constraints has already been incorporated into the objective function we need not consider it further and the problem becomes:

$$\text{Min}\left\{C_T\frac{\pi D^2}{2} + C_S\frac{4V}{D} = \text{cost}\right\} \text{ with } D \ge 0. \tag{9.5}$$

If the optimum value of D is positive, we can find the minimum by differentiating the cost with respect to D and setting this to zero.

$$\frac{d(\text{cost})}{dD} = C_T\pi D - \frac{4VC_S}{D^2} = 0 \tag{9.6}$$

Solving for variable D leads to the expression below with L obtained from the volume specification:

$$D = \left(\frac{4V}{\pi}\frac{C_S}{C_T}\right)^{1/3} \qquad L = \left(\frac{4V}{\pi}\right)^{1/3}\left(\frac{C_T}{C_S}\right)^{2/3} \tag{9.7}$$

Moreover, the aspect ratio for the cylinder can be expressed in a compact form: $L/D = C_T/C_S$. If we further examine the cost function, we see that:

$$d^2(\text{cost})/dD^2 = C_T\pi + 8\,V\,C_S/D^3 > 0, \text{ for } D > 0, \tag{9.8}$$

and by the definitions in Appendix A, this function is convex over the (open) feasible region for D. As a result, the solution to this NLP is a global one and no other (local) solutions exist.

In the next example, however, we have multiple solutions due to nonconvexity.

EXAMPLE 9.2 Minimize Packing Dimensions

Consider three cylindrical objects of equal height but with three different radii, as shown in Figure 9.1 below. What is the box with the smallest perimeter that will contain these three cylinders? Formulate and analyze this nonlinear programming problem.

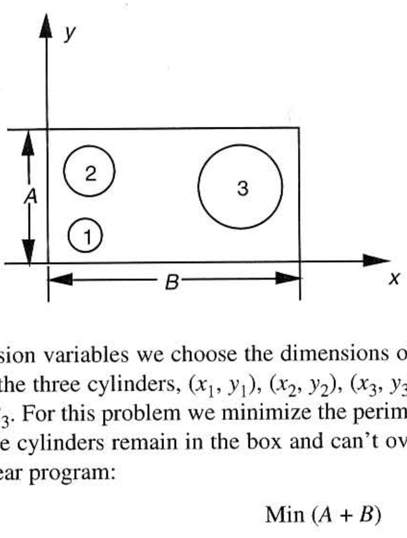

FIGURE 9.1 Illustration of Example 9.2

As decision variables we choose the dimensions of the box, A, B, and the coordinates for the centers of the three cylinders, (x_1, y_1), (x_2, y_2), (x_3, y_3). As specified parameters we have the radii, R_1, R_2, R_3. For this problem we minimize the perimeter $2(A + B)$ and include as constraints the fact that the cylinders remain in the box and can't overlap. As a result we formulate the following nonlinear program:

$$\text{Min } (A + B) \tag{9.9}$$

$$\text{in box} \begin{cases} x_1,\ y_1 \ge R_1 & x_1 \le B - R_1,\ y_1 \le A - R_1 \\ x_2,\ y_2 \ge R_2 & x_2 \le B - R_2,\ y_2 \le A - R_2 \\ x_3,\ y_3 \ge R_3 & x_3 \le B - R_3,\ y_3 \le A - R_3 \end{cases}$$

$$\text{no overlaps} \begin{cases} (x_1 - x_2)^2 + (y_1 - y_2)^2 \ge (R_1 + R_2)^2 \\ (x_1 - x_3)^2 + (y_1 - y_3)^2 \ge (R_1 + R_3)^2 \\ (x_2 - x_3)^2 + (y_2 - y_3)^2 \ge (R_2 + R_3)^2 \end{cases}$$

$$x_1, x_2, x_3, y_1, y_2, y_3, A, B \geq 0$$

Note that the objective function and the "in box" constraints are linear, and hence, convex. Similarly, the variable bounds are convex as well. The nonconvexities are observed in the nonlinear inequality constraints and this can be verified using the properties in Appendix A (see Exercise 9.1). Because convexity conditions are not satisfied, there is no guarantee of a unique global solution. Indeed, we can imagine intuitively the existence of multiple solutions to this NLP, as follows:

- Find a solution and observe an equivalent solution by turning the box by 90°.
- Use a random arrangement for the cylinders and manually shrink the walls of the box. The solution depends on the initial positions of the cylinders.

Consequently, we see that this problem has many local solutions. This is due to a nonconvex feasible region.

These two examples raise some interesting questions that will be explored next. First, what are the conditions that characterize even a local solution to a nonlinear program? In the first example, once L was eliminated, the constraints became unimportant. On the other hand, in the second example, the NLP solution was completely defined by the constraints. At the solution, these inequality constraints were satisfied as equations and were therefore considered to be *active*. In the remainder of this section we will present the *Kuhn Tucker optimality conditions* to define locally optimal solutions.

Second, the search for NLP solutions is guided by determining the correct active set of constraints and the solution of equations that represent the optimality conditions. In the first example this task was easy as no active constraints were considered, and because the optimal solution could be found analytically from Eq. (9.6). In the second example, we have yet to consider these tasks. These search strategies will be considered when we develop an NLP algorithm in the next section.

9.2.1 Optimality Conditions for Nonlinear Programming

In the remainder of this section we briefly present and discuss the optimality conditions for solution of the nonlinear programming problem (1). These are derived in Appendix A and are presented in detail below. Before presenting these properties, we first consider an intuitive explanation of the optimality conditions.

Consider the contour plot of $f(x)$ in two dimensions as shown in Figure 9.2. By inspection we see that the minimum point is given by x^*. If we consider this plot as a (smooth) valley, then a "ball" rolling in this valley will stop at x^*, the lowest point. At this stationary point we have a zero gradient, $\nabla f(x^*) = 0$, and the second derivatives reveal positive curvature of $f(x)$. In other words, if we move the ball away from x^* in any direction, it will roll back.

Now if we introduce two inequality constraints, $g_1(x) \leq 0$ and $g_2(x) \leq 0$, into the minimization problem, we can visualize this as imposing two "fences" in the valley, as

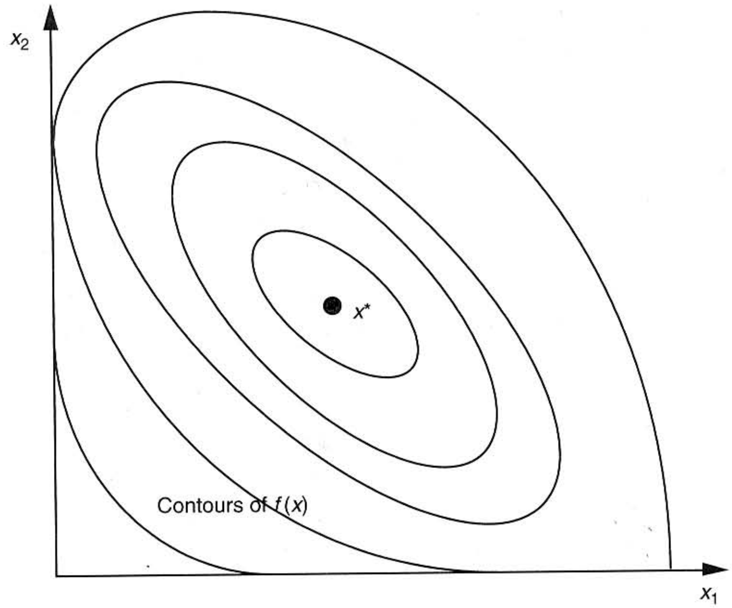

FIGURE 9.2 Contour plot for unconstrained minimum.

shown in Figure 9.3. Again, a ball rolling in the valley within the fences will roll to the lowest allowable point. However, if x^* is at the boundary of a constraint (e.g., $g_1(x^*) = 0$), then this inequality constraint is *active,* the ball is pinned at the fence and we no longer have $\nabla f(x^*) = 0$. Instead, we see that the ball remains stationary because of a balance of "forces": the force of "gravity" ($-\nabla f(x^*)$) and the "normal force" exerted on the ball by the fence ($-\nabla g_1(x^*)$). Also, in Figure 9.3 note that the constraint $g_2(x) \leq 0$ is inactive at x^* and does not participate in this "force balance." In addition to the balance of forces, we expect positive curvature *along the active constraint*; that is, if we move the ball from x^* in any direction along the fence, it will roll back.

Finally, we introduce an equality constraint, $h(x) = 0$, into the problem and we can visualize this as introducing a "rail" into the valley, as shown in Figure 9.4. Now a ball rolling on the rail and within the fence will also stop at the lowest point, x^*. This point will also be characterized by a balance of "forces": the force of "gravity" ($-\nabla f(x^*)$), the "normal force" exerted on the ball by the fence ($-\nabla g_1(x^*)$), and the "normal force" exerted on the ball by the rail ($-\nabla h(x^*)$). In addition to this balance of forces, we expect positive curvature *along the active constraints.* However, in Figure 9.4, we no longer have allowable directions that remain on the active constraints. Instead, the ball remains stationary at the intersection of the rail and the fence—and this condition is sufficient for optimality.

We now generalize these concepts and develop the optimality conditions for constrained minimization. These optimality conditions are referred to as the Kuhn Tucker

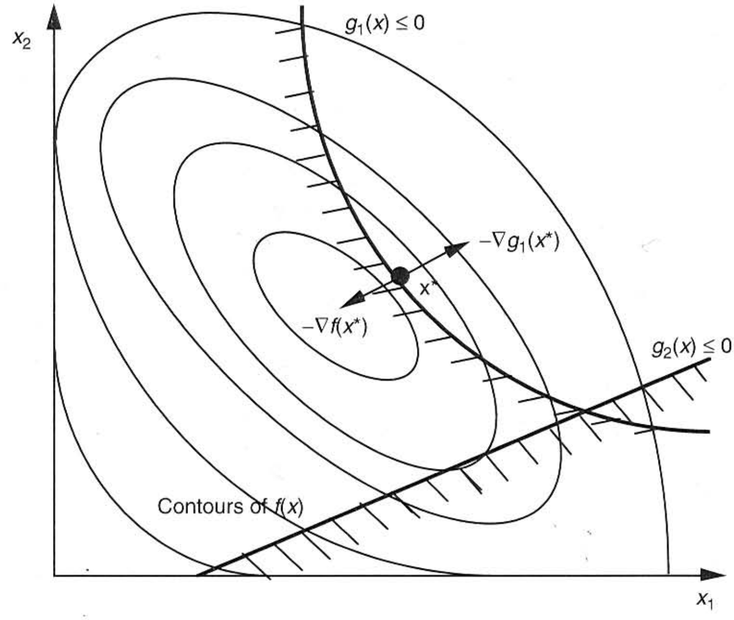

FIGURE 9.3 Constrained minimization with inequalities.

(KT) conditions or Karush Kuhn Tucker (KKT) conditions and were developed independently by Karush (1939) and Kuhn and Tucker (1951). For convenience of notation we define a Lagrange function as:

$$L(x, \mu, \lambda) = f(x) + g(x)^T\mu + h(x)^T\lambda = 0 \tag{9.10}$$

Here the vectors μ and λ act as "weights" for balancing the "forces" shown in Figure 9.4; μ and λ are referred to as *dual variables* or *Kuhn Tucker multipliers*. They are also called *shadow prices* in operations research literature.

The solution of the NLP (9.1) satisfies the following first-order Kuhn Tucker conditions. These conditions are *necessary* for optimality.

1. Linear dependence of gradients ("balance of forces" in Figure 9.4)

$$\nabla L(x^*, \mu^*, \lambda^*) = \nabla f(x^*) + \nabla g(x^*)\mu^* + \nabla h(x^*)\lambda^* = 0 \tag{9.11}$$

2. Feasibility of NLP solution (within the fences and on the rail in Figure 9.4)

$$g(x^*) \leq 0, \; h(x^*) = 0 \tag{9.12}$$

3. Complementarity condition; either $\mu_i^* = 0$ or $g_i(x^*) = 0$ (either at the fence boundary or not in Figure 9.4)

$$\mu^{*T} g(x^*) = 0 \tag{9.13}$$

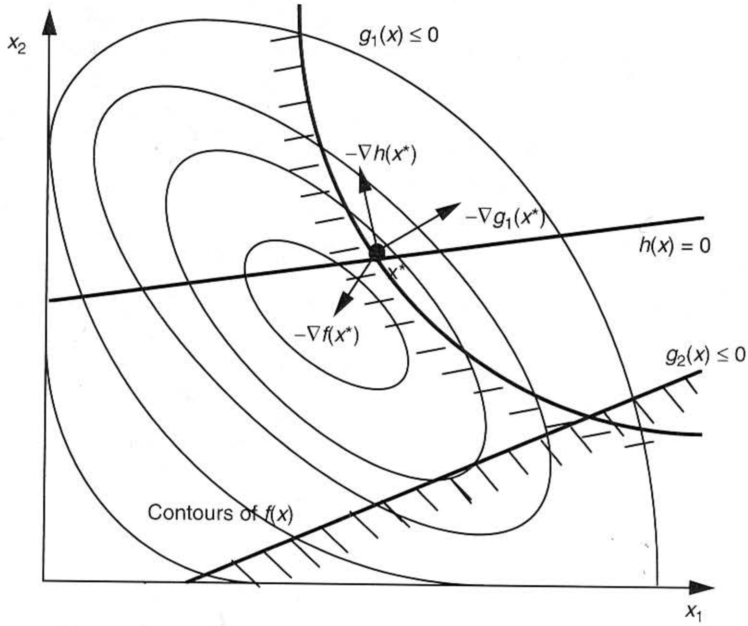

FIGURE 9.4 Constrained minimization with inequalities and equalities.

4. Nonnegativity of inequality constraint multipliers (normal force from "fence" can only act in one direction)

$$\mu^* \geq 0 \tag{9.14}$$

5. Constraint qualification:
   Active constraint gradients, i.e.:

$$[\nabla g_A(x^*) \mid \nabla h(x^*)] \text{ for } i \in A, A = \{i \mid g_i(x^*) = 0\}$$

   must be linearly independent.

The first Kuhn Tucker condition Eq. (9.11) describes linear dependence of the gradients of the objective and constraint functions and is derived in Appendix A. The second condition Eq. (9.12) requires that the solution of the NLP, x^*, satisfy all the constraints. The third and fourth conditions Eqs. (9.13, 9.14) relate to complementarity. Here either inequality constraint i is inactive ($g_i(x^*) < 0$) and the corresponding multiplier is zero (i.e., the constraint is ignored in the KT conditions), or, if the constraint is active ($g_i(x^*) = 0$), μ_i can be positive. Finally, in order for a local NLP solution to satisfy the KT conditions, an additional constraint qualification is required. Constraint qualifications take several forms (see Fletcher, 1987), and the one most frequently invoked is that the gradients of the active constraints be linearly independent.

These conditions are only necessary, however, and additional conditions are needed to ensure that x^* is a local solution. So far, the first order conditions define x^* only as a stationary point that satisfies the constraints. For instance, in Example 9.1, the KT conditions Eqs. (9.11–9.14) correspond to setting the gradient of the objective function to zero. To confirm a local optimum for this example, second derivatives have to be evaluated and checked to be positive (or at least nonnegative).

For a multivariable problem, the second derivatives are evaluated in terms of a *Hessian matrix* of a given function. For instance, the Hessian matrix of the objective function, $\nabla_{xx}f(x)$, is made up of elements: $\{\nabla_{xx}f(x)\}_{ij} = \partial^2 f/\partial x_i \partial x_j$. Also, since $\partial^2 f/\partial x_j \partial x_i = \partial^2 f/\partial x_i \partial x_j$, we have $\{\nabla_{xx}f(x)\}_{ij} = \{\nabla_{xx}f(x)\}_{ji}$ and the Hessian matrix is symmetric. Moreover, positive curvature for the contour surface can be evaluated based on the Hessian matrix. For instance, the objective function in Figure 9.2 has positive curvature at x^* if its Hessian matrix is positive definite, i.e.:

$$p^T\nabla_{xx}f(x^*)\,p > 0 \qquad \text{for all vectors } p \neq 0$$

or positive semidefinite:

$$p^T\nabla_{xx}f(x^*)\,p \geq 0 \qquad \text{for all vectors } p \neq 0.$$

For the constrained NLP problem (1), second order conditions are defined using the Hessian matrix of the Lagrange function and by defining nonzero *allowable directions* for the optimization variables based on the active constraints. Starting from the solution x^*, the allowable directions, p, satisfy the active constraints as equalities and therefore remain in the feasible region. Because, the change in x along this direction can be arbitrarily small, these directions must also satisfy linearizations of these constraints and are therefore defined by:

$$\nabla h(x^*)^T p = 0 \tag{9.15}$$

$$\nabla g_i(x^*)^T p = 0 \text{ for } i \in A,\ A = \{i \mid g_i(x^*) = 0\}$$

The sufficient (necessary) *second order conditions* require positive (nonnegative) curvature of the Lagrange function in these allowable or "constrained" directions, p. Using the second derivative matrix to define this curvature we express these conditions as:

$$\begin{aligned} p^T \nabla_{xx}L(x^*, \mu^*, \lambda^*)\,p &> 0 \text{ (sufficient condition)} \\ p^T \nabla_{xx}L(x^*, \mu^*, \lambda^*)\,p &\geq 0 \text{ (necessary condition)} \end{aligned} \tag{9.16}$$

for all of the allowable directions, p. These second order conditions are also presented in more detail in Appendix A.

EXAMPLE 9.3 Application of Kuhn Tucker Conditions

To illustrate these Kuhn Tucker conditions, we consider two simple examples represented in Figure 9.5.

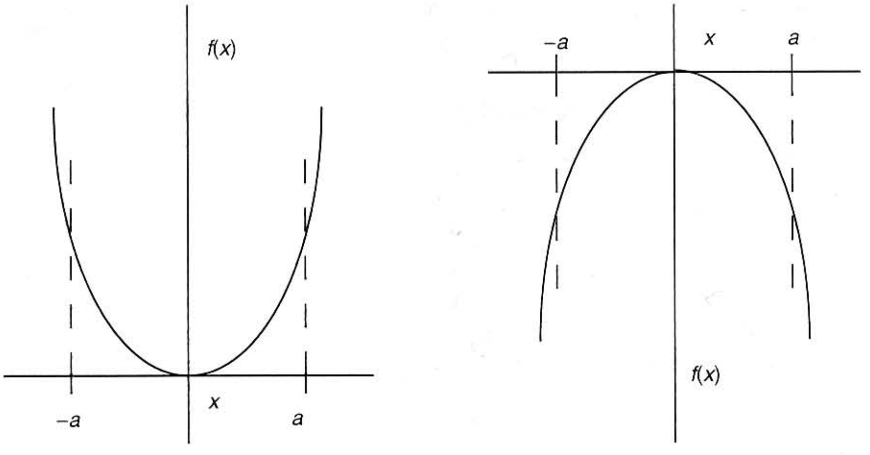

FIGURE 9.5 Illustration of Kuhn Tucker conditions for Example 9.3

First, we consider the single variable problem:

$$\text{Min } x^2 \quad s.t. \ -a \le x \le a, \text{ where } a > 0 \tag{9.17}$$

where $x^* = 0$ is seen by inspection. The Lagrange function for this problem can be written as:

$$L(x, \mu) = x^2 + \mu_1 (x - a) + \mu_2 (-a - x) \tag{9.18}$$

with the first order Kuhn Tucker conditions Eqs. (9.11–9.14) given by:

$$\begin{aligned} &\nabla L(x, \mu) = 2\,x + \mu_1 - \mu_2 = 0 \\ &\mu_1 (x - a) = 0 \quad \mu_2 (-a - x) = 0 \\ &-a \le x \le a \qquad \mu_1, \mu_2 \ge 0 \end{aligned} \tag{9.19}$$

To satisfy the first order conditions Eq. (9.19) we consider three cases: $\mu_1 = \mu_2 = 0$; $\mu_1 > 0$, $\mu_2 = 0$; or $\mu_1 = 0$, $\mu_2 > 0$. Note that the case $\mu_1 > 0$, $\mu_2 > 0$ cannot exist for $a > 0$ (Why?).Satisfying these conditions requires the evaluation of three candidate solutions:

- Upper bound is active, $x = a$, $\mu_1 = -2a$, $\mu_2 = 0$
- Lower bound is active, $x = -a$, $\mu_2 = -2a$, $\mu_1 = 0$
- Neither bound is active, $\mu_2 = 0$, $\mu_1 = 0$, $x = 0$

Clearly only the last case satisfies these conditions because the first two lead to negative values for μ_1 or μ_2. If we evaluate the second order conditions Eq. (9.16) we have allowable directions $p = \Delta x$ with $\Delta x > 0$ and $\Delta x < 0$. Also, we have

$$\begin{aligned} &\nabla_{xx} L\,(x^*, \mu^*, \lambda^*) = 2 > 0 \quad \text{and} \\ &p^T\,\nabla_{xx} L\,(x^*, \mu^*, \lambda^*)\,p = 2\,\Delta x^2 > 0 \end{aligned} \tag{9.20}$$

for all allowable directions. Therefore, the solution $x^* = 0$ satisfies both the sufficient first and second order Kuhn Tucker conditions for a local minimum.

We now consider an interesting variation on this example. As seen in Figure 9.5, suppose we change the sign on the objective function and solve:

$$\text{Min } -x^2 \qquad s.t. \; -a \le x \le a, \text{ where } a > 0. \tag{9.21}$$

Here the solution, $x^* = a$ or $-a$, is seen by inspection. The Lagrange function for this problem is now written as:

$$L(x, \mu) = -x^2 + \mu_1 (x - a) + \mu_2 (-a - x) \tag{9.22}$$

with the first order Kuhn Tucker conditions given by:

$$\begin{aligned} &\nabla L(x, \mu) = -2\,x + \mu_1 - \mu_2 = 0 \\ &\mu_1 (x - a) = 0 \qquad \mu_2 (-a - x) = 0 \\ &-a \le x \le a \qquad\quad \mu_1, \mu_2 \ge 0 \end{aligned} \tag{9.23}$$

Again, satisfying conditions (9.23) requires the evaluation of three candidate solutions, depending on $\mu_1 = \mu_2 = 0$; $\mu_1 > 0, \mu_2 = 0$; or $\mu_1 = 0, \mu_2 > 0$:

- Upper bound is active, $x = a$, $\mu_1 = 2a$, $\mu_2 = 0$
- Lower bound is active, $x = -a$, $\mu_2 = 2a$, $\mu_1 = 0$
- Neither bound is active, $\mu_2 = 0$, $\mu_1 = 0$, $x = 0$

and all three cases satisfy the first order conditions. We now need to check the second order conditions to discriminate among these points. If we evaluate the second order conditions (16) at $x = 0$, we realize allowable directions $p = \Delta x > 0$ and $-\Delta x$ and we have:

$$p^T \nabla_{xx} L\,(x, \mu, \lambda)\, p = -2\,\Delta x^2 < 0. \tag{9.24}$$

This point does not satisfy the second order conditions. In the other two cases, we invoke a subtle concept. *For $x = a$ or $x = -a$, we require the allowable direction to satisfy the active constraints exactly. Here, any point along the allowable direction, x^* must remain at its bound.* For this problem, however, there are no nonzero allowable directions that satisfy this condition. Consequently, the solution x^* is defined entirely by the active constraint. The condition:

$$p^T \nabla_{xx} L\,(x^*, \mu^*, \lambda^*)\, p > 0 \tag{9.25}$$

for all allowable directions, is *vacuously* satisfied—because there are *no* allowable directions.

The first and second order Kuhn Tucker conditions provide a useful tool for identifying local solutions to nonlinear programs. (It should be noted, though, that because second derivatives are often not calculated in process optimization problems, second order conditions are rarely checked.) However, we still need efficient search strategies that locate points that satisfy these conditions. In the next section, we develop a nonlinear programming algorithm called Successive Quadratic Programming (SQP). For process optimization, this algorithm has some desirable features and it has been used widely in many process applications. Moreover, it has proved to be adaptable to several kinds of nonlinear programming problems.