Journal of Global Optimization **7**: 127–142, 1995.

Nonconvex Optimization over a Polytope Using Generalized Capacity Improvement

B. W. LAMAR
Department of Management, University of Canterbury, Christchurch, New Zealand

(Received: 1 April 1995; accepted: 1 June 1995)

Abstract. Capacity improvement involves the reduction of the size of the feasible region of an optimization problem without 'cutting-off' the optimal solution point(s). Capacity improvement can be used in a branch-and-bound procedure to produce tighter relaxations to subproblems in the enumeration tree. Previous capacity improvement work has concentrated on tightening the simple lower and upper bounds on variables. In this paper, the capacity improvement procedure is generalized to apply to *all* constraints that form the feasible region. For the minimization of a separable concave function over a bounded polytope, the method of calculating the capacity improvement parameters is very straightforward. Computational results for fixed-charge and quadratic concave minimization problems demonstrate the effectiveness of this procedure.

Key words: Capacity improvement, domain reduction, nonconvex optimization, branch-and-bound.

1. Introduction

Capacity improvement can be classified as a 'domain reduction' technique. Domain reduction involves the 'shrinking' of the feasible region of an optimization problem while, at the same time, ensuring that the tighter region thus produced still contains the optimal solution point(s) to the original problem. Techniques such as variable 'pegging' [14] for fixed-charge problems or Tuy [26] cuts for general concave minimization problems can be classified as domain reduction techniques. A distinguishing feature of capacity improvement, however, is that it does not add additional constraints to the problem. Instead, it tightens the feasible region by systematically altering the right-hand-side coefficients of the constraints of the problem. In this way, tighter relaxations to the original problem can be produced without increasing the complexity of these relaxations. Additional domain reduction techniques are discussed in [2, 4, 5, 6, 10, 11, 15, 16, 18, 23, 24, 25].

To illustrate the effect of the generalized capacity improvement procedure presented in this paper, consider problem P, the minimization of a separable concave function over a bounded polytope. Problem P is specified as

$$\text{Problem } P: \quad \min \phi(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X = G \cap H,$$

where $\underline{x} = (\ldots, x_j, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ is the decision variable vector; $G = \{\underline{x} : \underline{A}\,\underline{x} = \underline{b}\}$ is a polytope with $\underline{A} \in \mathbf{R}^{m \times n}$ and $\underline{b} = (\ldots, b_i, \ldots)^{\mathrm{T}} \in \mathbf{R}^m$; $H = \{\underline{x} : \underline{l} \leq \underline{x} \leq \underline{u}\}$ is a hyper-rectangle with $\underline{l} = (\ldots, l_j, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ and $\underline{u} = (\ldots, u_j, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$;

and $\phi(\underline{x}) = \Sigma_{j=1}^{n}\phi_j(x_j)$ is the objective function with each $\phi_j(x_j)$ a concave function over $l_j \le x_j \le u_j$. Let $\underline{x}^* = (\ldots, x_j^*, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ denote an optimal solution vector of problem P. Also, throughout this paper, for any problem $\bullet$, let $v[\bullet]$ denote the optimal objective function value of $\bullet$, let $lb[\bullet]$ be a lower bound to $v[\bullet]$, and let $ub[\bullet]$ be an upper bound to $v[\bullet]$. Thus, $v[P] = \phi(\underline{x}^*)$.

Now consider the set $\tilde{X} \subseteq \mathbf{R}^n$. If $\underline{x}^* \in \tilde{X}$, then $\tilde{X}$ is termed a 'valid' region for problem P. If, in addition to being valid, we have $\tilde{X} \subset X$, then $\tilde{X}$ is called an 'improved' region for problem P. Domain reduction techniques, such as capacity improvement, produce improved feasible regions. Previous work on capacity improvement has focused on generating an improved region $\tilde{X}$ by producing improved lower and upper variable bounds, $\tilde{\underline{l}}$ and $\tilde{\underline{u}}$. This is the approach taken in [15, 16, 18, 23, 25]. In this paper, we extend the ability of the capacity improvement procedure to generate an improved region $\tilde{X}$ by producing improved bounds $\tilde{\underline{l}}$ and $\tilde{\underline{u}}$ as well as an improved right-hand-side coefficient vector $\tilde{\underline{b}}$. This extension to the capacity improvement procedure is referred to as 'generalized capacity improvement'. Generalized capacity improvement was first explored in [17]. It has also been used successfully in [24].

Most methods for solving problem P involve extreme point ranking, cutting plane, or branch-and-bound techniques. These methods are surveyed in [9, 12, 13, 19, 20, 21, 22]. In the branch-and-bound method, the rectangle H is partitioned into successively smaller rectangles producing a series of subproblems of problem P. Let Q be a generic subproblem of P and let X_Q denote the feasible region of Q. Domain reduction techniques, such as generalized capacity improvement, can be applied to each subproblem Q to produce an improved region $\tilde{X}_Q$. Note that since $\tilde{X}_Q \subset X_Q$, a relaxation based on $\tilde{X}_Q$ will, in general, produce a lower bound to $v[Q]$ that is tighter than one based on X_Q. This tighter lower bound to $v[Q]$, in turn, facilitates the fathoming of subproblem Q. Branch-and-bound algorithms that incorporate domain reduction have been aptly labeled by Ryoo and Sahinidis [23] as 'branch-and-reduce' methods.

This paper is organized as follows. Section 2 outlines a branch-and-bound algorithm that uses generalized capacity improvement for domain reduction of the subproblems. Section 3 describes a straightforward method of computing the parameters used in the generalized capacity improvement procedure. Section 4 reports on the computational performance of this technique for a class of fixed-charge and quadratic concave minimization problems. Finally, Section 5 summarizes the paper and discusses future work in this area.

2. Solution Method

This section is divided into two parts. The first part summarizes the traditional branch-and-bound method for solving problem P using rectangular partitioning (see, for example, [22]). This part also introduces notation. The second part describes how generalized capacity improvement can be incorporated within the traditional branch-and-bound algorithm.

2.1. Branch-and-Bound Procedure

Rather than solving P directly, the branch-and-bound procedure partitions the rectangle H into smaller rectangles. This creates a series of 'subproblems' (i.e., 'descendants') of problem P. Let problem Q denote the subproblem of P currently under consideration. Specifically,

$$\text{Problem } Q: \quad \min \phi(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X_Q = G_Q \cap H_Q,$$

where $G_Q = \{\underline{x} : \underline{A}\,\underline{x} = \underline{b}_Q\}$ and $H_Q = \{\underline{x} : \underline{l}_Q \leq \underline{x} \leq \underline{u}_Q\}$ with $\underline{b}_Q = (\ldots, b_{Qi}, \ldots)^{\mathrm{T}} \in \mathbf{R}^m, \underline{l}_Q = (\ldots, l_{Qj}, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ and $\underline{u}_Q = (\ldots, u_{Qj}, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ such that $G_Q \subseteq G$ and $H_Q \subseteq H$. In the traditional branch-and-bound procedure using rectangular partitioning, $\underline{b}_Q \equiv \underline{b}$. Later, when capacity improvement is discussed, $\underline{b}_Q$ will be distinct from $\underline{b}$.

Problem Q (and its descendants) can be eliminated from further consideration if the following fathoming criterion, denoted criterion (*F*), is satisfied:

(*F*) $lb[Q] \geq ub[P]$

Typically, the value of $ub[P]$ is taken as the objective function value of the current 'incumbent' solution of P (i.e., best feasible solution to P found so far); and the value of $lb[Q]$ is obtained by solving a 'relaxation' of Q.

In general, a relaxation of Q can be any problem such that the following two properties, denoted (*R1*) and (*R2*), are satisfied [7]:

(*R1*) The feasible region of the relaxation contains or equals X_Q;

(*R2*) The objective function value of the relaxation at each point $\underline{x} \in X_Q$ is less than or equal to $\phi(\underline{x})$.

For our purposes, it is important to distinguish between properties (*R1*) and (*R2*). Therefore, we define a 'feasibility relaxation' of Q as any problem whose feasible region satisfies property (*R1*) but whose objective function *equals* $\phi(\underline{x})$ for all $\underline{x} \in X_Q$. Similarly, we define an 'objective value relaxation' of Q as any problem whose objective function satisfies property (*R2*) but whose feasible region *equals* X_Q.

Let $\overline{Q}$ be an objective value relaxation of Q. Specifically,

$$\text{Problem } \overline{Q}: \quad \min \overline{\phi}_Q(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X_Q = G_Q \cap H_Q,$$

where $\overline{\phi}_Q(\underline{x}) = \Sigma_{j=1}^{n} \overline{\phi}_{Qj}(x_j)$ is the convex (in our case, affine) lower envelope of $\phi(\underline{x})$ on H_Q. Because X_Q is a polytope and $\overline{\phi}_Q(\underline{x})$ is affine, problem $\overline{Q}$ is a linear program which can be solved very efficiently. Since $\overline{Q}$ is a relaxation of Q, we define $v[\overline{Q}]$ as the 'relaxation lower bound' to $v[Q]$.

If the fathoming criterion (F) is satisfied using the relaxation lower bound, then no further evaluation of problem Q is required and another subproblem of P can be selected and evaluated as the current subproblem. On the other hand, if criterion (F) is not satisfied, then one of the following two 'branching' actions, denoted ($B1$) and ($B2$), must be taken by the branch-and-bound procedure:

($B1$) 'Persist' at the current subproblem by evaluating a tighter relaxation of Q;

($B2$) 'Separate' the current subproblem by partitioning the rectangle H_Q into two smaller rectangles thereby replacing Q with two new subproblems of P.

Following action ($B1$), the branch-and-bound procedure will again attempt to fathom problem Q using a tighter value of $lb[Q]$ whereas following action ($B2$), a new subproblem will be selected and the branch-and-bound procedure will attempt to fathom the new subproblem. In either case, the branch-and-bound procedure will continue until all subproblems have been fathomed using criterion (F). Once all the subproblems of P have been fathomed, the branch-and-bound procedure terminates and the current incumbent solution to P is identified as an optimal solution to P.

2.2. Generalized Capacity Improvement

Generalized capacity improvement is one method of taking action ($B1$) (i.e., 'persisting' at the current subproblem, Q). In order to explain the concept, we define another problem, denoted Q^*, whose feasible region, denoted X_Q^*, is the (possibly empty) subset of X_Q in which the value of $\phi(\underline{x})$is less than or equal to $ub[P]$ (the current upper bound to $v[P]$). Specifically,

$$\text{Problem } Q^*: \quad \min \phi(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X_Q^* = G_Q \cap H_Q \cap \{\underline{x} : \phi(\underline{x}) \leq ub[P]\}.$$

Thakur [25] refers to problem Q^* as a 'contraction' of problem Q. Note that if $ub[P] \geq v[Q]$, then $v[Q^*] = v[Q]$. On the other hand, if $ub[P] < v[Q]$, then Q^* is infeasible so X_Q^* is empty and $v[Q^*] = +\infty$.

In addition, let Q^r for $r = 0, 1, 2, \ldots,$ be a family of successively tighter feasibility relaxations of Q^*. Specifically,

$$\text{Problem } Q^r: \quad \min \phi(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X_Q^r = G_Q^r \cap H_Q^r,$$

where $G_Q^r = \{\underline{x} : \underline{A}\,\underline{x} = \underline{b}_Q^r\}$ and $H_Q^r = \{\underline{x} : \underline{l}_Q^r \leq \underline{x} \leq \underline{u}_Q^r\}$ with $\underline{b}_Q^r = (\ldots, b_{Qi}^r, \ldots)^{\mathrm{T}} \in \mathbf{R}^m, \underline{l}_Q^r = (\ldots, l_{Qj}^r, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ and $\underline{u}_Q^r = (\ldots, u_{Qj}^r, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ such that for each r we have $X_Q^r \supseteq X_Q^{r+1} \supseteq X_Q^*$. For $r = 0$, we set $G_Q^0 = G_Q$ and $H_Q^0 = H_Q$ so that $X_Q^0 = X_Q$ and problem Q^0 is the same as problem Q. Note that if $ub[P] \geq v[Q]$, then for each r we have $v[Q^r] = v[Q^{r+1}] = v[Q^*] = v[Q]$.

Next, we define problem $\overline{Q}^r$ to be an objective value relaxation of problem Q^r. Specifically,

$$\text{Problem } \overline{Q}^r: \quad \min \overline{\phi}_Q^r(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in X_Q^r = G_Q^r \cap H_Q^r,$$

where $\overline{\phi}_Q^r(\underline{x}) = \Sigma_{j=1}^n \overline{\phi}_{Qj}^r(x_j)$ is the convex (in our case, affine) lower envelope of $\phi(\underline{x})$ on H_Q^r. Problem $\overline{Q}^r$ is a linear program which can be solved very efficiently. Note that, because $\overline{Q}^r$ is a relaxation of Q^r, we have $v[\overline{Q}^r] \leq v[Q^r]$. Furthermore, if $ub[P] \geq v[Q]$, then for each r, we have $v[Q^r] = v[Q]$, so $v[\overline{Q}^r]$ is a lower bound to $v[Q]$. On the other hand, if $ub[P] < v[Q]$, then $ub[P]$ is itself a lower bound to $v[Q]$. This means that either $v[\overline{Q}^r]$ or $ub[P]$ (or possibly both) is a lower bound to $v[Q]$.

Therefore, for each r, we let CI_Q^r denote the 'capacity improvement lower bound' to $v[Q]$, where CI_Q^r is defined as

$$CI_Q^r = \min \{ub[P], v[\overline{Q}^r]\}. \tag{1}$$

The generalized capacity improvement procedure produces a sequence of non-decreasing lower bounds each of which is as least as tight as the relaxation lower bound $v[\overline{Q}]$. If, for any given r, criterion (*F*) is satisfied using the capacity improvement lower bound, then problem Q can be fathomed. Otherwise, either actions (*B1*) or (*B2*) must be taken. If action (*B1*) is selected, then the branch-and-bound procedure will attempt to produce a tighter lower bound to $v[Q]$ by solving problem $\overline{Q}^{r+1}$ and computing CI_Q^{r+1}. On the other hand, if action (*B2*) is chosen, then rectangle H_Q^r will be partitioned into two smaller rectangles.

The next section describes how CI_Q^r is calculated by specifying the parameters in problem $\overline{Q}^r$.

3. Parameter Determination

Throughout this section we assume that we are focusing on the rth capacity improvement iteration of subproblem Q. Therefore, we omit the subscript 'Q' and the superscript 'r' throughout this section. Also, when we are referring to the $(r+1)$-st capacity improvement iteration, we use the superscript '1' rather than the superscript '$r+1$' throughout this section. Thus, for instance, in the rth iteration, (1) is given as

$$CI = \min \{ub[P], v[\overline{Q}]\} \tag{2}$$

and, in the $(r+1)$-st iteration, (1) is specified as

$$CI^1 = \min \{ub[P], v[\overline{Q}^1]\}. \tag{3}$$

We assume that, in the rth iteration, the solution to the linear program $\overline{Q}$ is available. The solution to problem $\overline{Q}$ (in the rth iteration) is used to determine the 'improved' parameter vectors, $\underline{b}^1 = (\ldots, b_i^1, \ldots)^{\mathrm{T}} \in \mathbf{R}^m$, $\underline{l}^1 = (\ldots, l_j^1, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ and $\underline{u}^1 = (\ldots, u_j^1, \ldots)^{\mathrm{T}} \in \mathbf{R}^n$ (in the $(r+1)$-st iteration). Once the improved parameter vectors have been determined, problem $\overline{Q}^1$ can be solved and the value of $v[\overline{Q}^1]$ can be determined. This value, in turn, is used to calculate CI^1 using (3). Thus, using this 'boot-strap' method, the capacity improvement lower bound can be computed for *any* iteration number, r.

This section is divided into four subsections. The first two subsections describe the method for computing the improved right-hand-side vector $\underline{b}^1$; and, for completeness, the last two subsections review the method for computing the improved lower and upper bound vectors, $\underline{l}^1$ and $\underline{u}^1$.

3.1. Greater-than-or-equal-to Constraint

Let $\underline{a}_i \in \mathbf{R}^n$ be a row vector denoting the ith row of constraint matrix $\underline{A}$. Then, the ith constraint of the polyhedral constraint set G of problem $\overline{Q}$ is given by $\underline{a}_i\underline{x} = b_i$. Suppose that this equality constraint was formed by introducing a surplus variable to a greater-than-or-equal-to constraint. Let $k(i)$ denote the index of the surplus variable for the ith constraint and let $\underline{e}_{k(i)} \in \mathbf{R}^n$ be a unit row vector (i.e., a vector of zeros except for a 1 in the $k(i)$th position). Then, the ith constraint of problem $\overline{Q}$ can also be expressed as $(\underline{a}_i + \underline{e}_{k(i)})\underline{x} \geq b_i$.

Now consider the effect of adding the *less*-than-or-equal-to constraint

$$\underline{d}_i\underline{x} \leq \theta \tag{4}$$

to problem $\overline{Q}$ where $\underline{d}_i = \underline{a}_i + \underline{e}_{k(i)}$ and θ is a scalar parameter. Note that constraint (4) is identical to the ith constraint of problem $\overline{Q}$ except for the right-hand-side coefficient and the direction of the inequality. Let problem $\overline{Q}$ augmented with constraint (4) be denoted by $\overline{Q} \mid \underline{d}_i\underline{x} \leq \theta$ and let $v[\overline{Q} \mid \underline{d}_i\underline{x} \leq \theta]$ denote the optimal objective function value of this augmented problem. Note that $v[\overline{Q} \mid \underline{d}_i\underline{x} \leq \theta]$ is a piecewise-linear, convex, nonincreasing function of θ and that constraint (4) is binding in the optimal solution to the augmented problem only if $\theta \leq b_i + \overline{x}_{k(i)}$ where $\overline{x}_{k(i)}$ is the optimal value of decision variable $x_{k(i)}$ in problem $\overline{Q}$.

In addition, let α_i denote the rate of increase in $v[\overline{Q} \mid \underline{d}_i\underline{x} \leq \theta]$ for an incremental decrease in θ below $b_i + \overline{x}_{k(i)}$. As shown in the Appendix, α_i can be computed directly from the solution to problem $\overline{Q}$. By construction, $\alpha_i \geq 0$.

Next, let $\partial(\theta)$ denote the line given by

$$\partial(\theta) = v[\overline{Q}] + \alpha_i(b_i + \overline{x}_{k(i)}) - \alpha_i\theta. \tag{5}$$

Observe that $\partial(\theta)$ is a tangent line to the function $v[\overline{Q} \mid \underline{d}_i\underline{x} \leq \theta]$. It has slope $-\alpha_i$ and passes through the point $(b_i + \overline{x}_{k(i)}, v[\overline{Q}])$. If α_i is strictly greater than zero, then we can solve explicitly for the value of θ such that $\partial(\theta)$ equals the upper bound, $ub[P]$. Letting $\tilde{b}_i$ denote the value of θ such that $\partial(\theta) = ub[P]$, we obtain

$$\tilde{b}_i = b_i + \overline{x}_{k(i)} - \frac{ub[P] - v[\overline{Q}]}{\alpha_i}. \tag{6}$$

We now specify b_i^1 , the improved right-hand-side coefficient for the (greater-than-or-equal-to) constraint i, as

$$b_i^1 = \begin{cases} \tilde{b}_i & \text{if } \alpha_i > 0 \text{ and } \tilde{b}_i > b_i \\ b_i & \text{otherwise} \end{cases}. \tag{7}$$

3.2. Less-than-or-equal-to Constraint

Now suppose that the ith constraint of $\overline{Q}$ was formed by adding a slack variable to a less-than-or-equal-to constraint. A procedure analogous to that given in Section 3.1 can be performed to determine the improved right-hand-side coefficient, b_i^1. Let $k(i)$ denote the index of the slack variable for the ith constraint. Then, the ith constraint of problem $\overline{Q}$ can be expressed as $(\underline{a}_i - \underline{e}_{k(i)})\underline{x} \le b_i$.

We now form the augmented problem by adding the *greater*-than-or-equal-to constraint

$$\hat{\underline{d}}_i\underline{x} \ge \theta \tag{8}$$

to problem $\overline{Q}$ where $\hat{\underline{d}}_i = \underline{a}_i - \underline{e}_{k(i)}$ and θ is a scalar parameter. Let problem $\overline{Q}$ augmented with constraint (8) be denoted by $\overline{Q} \mid \hat{\underline{d}}_i\underline{x} \ge \theta$ and let $v[\overline{Q} \mid \hat{\underline{d}}_i\underline{x} \ge \theta]$ denote the optimal objective function value of this augmented problem. For this case, note that $v[\overline{Q} \mid \hat{\underline{d}}_i\underline{x} \ge \theta]$ is a piecewise-linear, convex, nondecreasing function of θ and that constraint (8) is binding only if $\theta \ge b_i - \overline{x}_{k(i)}$.

Let β_i denote the rate of increase in $v[\overline{Q} \mid \hat{\underline{d}}_i\underline{x} \ge \theta]$ for an incremental increase in θ above $b_i - \overline{x}_{k(i)}$. By construction, $\beta_i \ge 0$. Now let $\partial(\theta)$ denote the tangent line given by

$$\partial(\theta) = v[\overline{Q}] - \beta_i(b_i - \overline{x}_{k(i)}) + \beta_i\theta. \tag{9}$$

If $\beta_i > 0$, then we can solve explicitly for the value of θ such that $\partial(\theta) = ub[P]$. Letting $\hat{b}_i$ denote this value, we obtain

$$\hat{b}_i = b_i - \overline{x}_{k(i)} + \frac{ub[P] - v[\overline{Q}]}{\beta_i}. \tag{10}$$

Then, the value of b_i^1 , the improved right-hand-side coefficient for the (less-than-or-equal-to) constraint i, is given by

$$b_i^1 = \begin{cases} \hat{b}_i & \text{if } \beta_i > 0 \text{ and } \hat{b}_i < b_i \\ b_i & \text{otherwise} \end{cases}. \tag{11}$$

3.3. Lower Bound

Let k be the index of one of the decision variables in problem $\overline{Q}$, let $\underline{e}_k \in \mathbf{R}^n$ be a unit vector, and let θ be a scalar. Consider the augmented problem formed by adding the constraint

$$\underline{e}_k\underline{x} \leq \theta \tag{12}$$

to problem $\overline{Q}$. Let γ_k denote the rate of increase in $v[\overline{Q} \mid \underline{e}_k\underline{x} \leq \theta]$ for an incremental decrease in θ below $\overline{x}_k$. By construction, $\gamma_k \geq 0$. Let $\partial(\theta)$ denote the tangent line given by

$$\partial(\theta) = v[\overline{Q}] + \gamma_k\overline{x}_k - \gamma_k\theta. \tag{13}$$

Letting $\tilde{l}_k$ denote the value of θ such that $\partial(\theta) = ub[P]$ for $\gamma_k > 0$, we obtain

$$\tilde{l}_k = \overline{x}_k - \frac{ub[P] - v[\overline{Q}]}{\gamma_k} \tag{14}$$

and the value of l_k^1 , the improved lower bound for variable x_k, is given by

$$l_k^1 = \begin{cases} \tilde{l}_k & \text{if } \gamma_k > 0 \text{ and } \tilde{l}_k > l_k \\ l_k & \text{otherwise} \end{cases}. \tag{15}$$

3.4. Upper Bound

Now consider the augmented problem formed by adding the greater-than-or-equal-to constraint

$$\underline{e}_k\underline{x} \geq \theta \tag{16}$$

to problem $\overline{Q}$ and let δ_k denote the rate of increase in $v[\overline{Q} \mid \underline{e}_k\underline{x} \geq \theta]$ for an incremental increase in θ above $\overline{x}_k$. By construction, $\delta_k \geq 0$. Let $\partial(\theta)$ denote the tangent line given by

$$\partial(\theta) = v[\overline{Q}] - \delta_k\overline{x}_k + \delta_k\theta \tag{17}$$

Letting $\tilde{u}_k$ denote the value of θ such that $\partial(\theta) = ub[P]$ for $\delta_k > 0$, we obtain

$$\tilde{u}_k = \overline{x}_k + \frac{ub[P] - v[\overline{Q}]}{\delta_k} \tag{18}$$

and the value of u_k^1, the improved upper bound for variable x_k, is given by

$$u_k^1 = \begin{cases} \tilde{u}_k & \text{if } \delta_k > 0 \text{ and } \tilde{u}_k < u_k \\ u_k & \text{otherwise} \end{cases}. \tag{19}$$

The next section examines the computational effectiveness of the generalized capacity improvement procedure.

4. Computational Performance

The solution method described in Sections 2 and 3 was applied to two types of nonconvex minimization problems. First, Gray's [8] fixed-charge test problems were solved. These test problems are referred to as problems FIXED-1A through FIXED-4C. Second, these same test problems were solved using a separable concave quadratic objective function in place of the fixed-charge objective function. The quadratic functions were generated using coefficients that were randomly chosen from a uniform distribution. The coefficient for the squared term in the objective function was forced to be negative to ensure that the functions were concave. The quadratic test problems are referred to as problems QUAD-1A through QUAD-4C. The characteristics of the 12 fixed-charage and the 12 quadratic test problems are summarized in Table I.

TABLE I. Characteristics of test problems

Problem name	Type of objective function	Number of constraints	Number of decision variables
FIXED-1A,1B,1C	Fixed-charge	10	9
FIXED-2A,2B,2C	Fixed-charge	18	15
FIXED-3A,3B,3C	Fixed-charge	30	19
FIXED-4A,4B,4C	Fixed-charge	28	30
QUAD-1A,1B,1C	Concave quadratic	10	9
QUAD-2A,2B,2C	Concave quadratic	18	15
QUAD-3A,3B,3C	Concave quadratic	30	19
QUAD-4A,4B,4C	Concave quadratic	28	30

Each of the 24 test problems was solved four times: first, using the traditional branch-and-bound approach without capacity improvement (see Section 2.1); second, using capacity improvement only for the lower and upper bound vectors ($\underline{l}_Q^r$ and $\underline{u}_Q^r$); third, using capacity improvement only for the right-hand-side coefficient vector ($\underline{b}_Q^r$); and fourth, using capacity improvement for the upper and lower bounds as well as the right-hand-side coefficients ($\underline{l}_Q^r$, $\underline{u}_Q^r$, and $\underline{b}_Q^r$). The solution methods were programmed in Fortran using the Lindo callable library to solve the objective value relaxations ($\overline{Q}^r$) for the subproblems. The problems were solved on a Micro-Source International microcomputer (comparable to an IBM-AT). For each problem and each solution method, the number of subproblems evaluated in the branch-and-bound enumeration tree and the total CPU time (in seconds) were recorded.

The results are shown in Tables II through V. Tables II and III give the number of subproblems solved for the fixed-charge and the concave quadratic test problems, respectively. Here, we see that the capacity improvement procedure had a pronounced effect. Compared to the traditional branch-and-bound method, the generalized capacity improvement procedure reduced the number of subproblems evaluated by over 50% for the fixed-charge problems, and by over 80% for the concave quadratic problems. In fact, when capacity improvement was applied to the quadratic problems, two-thirds of the test problems required only a single subproblem, indicating that these problems were solved at the 'root node' in the branch-and-bound enumeration tree. Tables II and III also show that a substantial proportion of the benefit of capacity improvement was due to the improvements of the lower and upper bounds rather than due to improvements in the right-hand-side coefficients.

TABLE II. Subproblems evaluated: Fixed-charge test problems

Problem name	Branch-and-bound solution method			
	Traditional	C.I. for $\underline{l}^r_Q, \underline{u}^r_Q$	C.I. for $\underline{b}^r_Q$	C.I. for $\underline{l}^r_Q, \underline{u}^r_Q, \underline{b}^r_Q$
FIXED-1A	65	31	49	29
FIXED-1B	47	25	29	13
FIXED-1C	129	87	95	47
FIXED-2A	299	173	297	141
FIXED-2B	243	179	203	121
FIXED-2C	2479	1447	2213	1303
FIXED-3A	1113	673	955	645
FIXED-3B	887	401	801	389
FIXED-3C	627	293	591	259
FIXED-4A	2089	1061	1747	807
FIXED-4B	1169	685	2135	1325
FIXED-4C	4471	2303	2561	1251
Average % improvement	0.00%	+44.04%	+10.27%	+51.56%

The achievements of the generalized capacity improvement to reduce the number of subproblems evaluated is tempered somewhat by the results relating to CPU time. This information is shown in Tables IV and V for the fixed-charge and quadratic test problems, respectively. We see that, on average, the generalized capacity improvement reduced the CPU time by about 2% for the fixed-charge problems and by over 17% for the quadratic problems. However, an even greater benefit was achieved when the capacity improvement was applied only to the lower and upper bounds. Here, the capacity improvement procedure reduced the CPU time by over 20% for the fixed-charge problems and by over 40% for the quadratic problems. In contrast, when the capacity improvement was used for only the

TABLE III. Subproblems evaluated: Concave quadratic test problems

Problem name	Branch-and-bound solution method			
	Traditional	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r$	C.I. for $\underline{b}_Q^r$	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r, \underline{b}_Q^r$
QUAD-1A	5	1	5	1
QUAD-1B	13	5	13	5
QUAD-1C	7	1	7	1
QUAD-2A	21	5	15	5
QUAD-2B	15	1	15	1
QUAD-2C	7	1	7	1
QUAD-3A	23	3	19	3
QUAD-3B	9	1	9	1
QUAD-3C	33	1	33	1
QUAD-4A	5	1	5	1
QUAD-4B	27	5	27	5
QUAD-4C	5	1	3	1
Average % improvement	0.00%	+83.07%	+7.16%	+83.07%

TABLE IV. CPU Time: Fixed-charge test problems

Problem name	Branch-and-bound solution method			
	Traditional	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r$	C.I. for $\underline{b}_Q^r$	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r, \underline{b}_Q^r$
FIXED-1A	20.76	13.07	19.83	14.17
FIXED-1B	16.42	12.75	12.58	8.35
FIXED-1C	43.61	39.60	43.28	30.37
FIXED-2A	162.14	125.39	218.80	146.98
FIXED-2B	130.94	115.29	154.46	117.48
FIXED-2C	1388.18	1110.53	1749.98	1381.71
FIXED-3A	785.44	755.50	916.98	958.89
FIXED-3B	629.83	466.98	768.29	592.87
FIXED-3C	415.45	307.42	544.80	387.83
FIXED-4A	2270.73	1512.76	3371.49	1984.02
FIXED-4B	1259.66	992.12	4128.75	3072.53
FIXED-4C	4933.24	3603.12	5348.26	3292.89
Average % improvement	0.00%	+21.70%	−33.74%	+1.98%

TABLE V. CPU time: Concave quadratic test problems

Problem name	Branch-and-bound solution method Traditional	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r$	C.I. for $\underline{b}_Q^r$	C.I. for $\underline{l}_Q^r, \underline{u}_Q^r, \underline{b}_Q^r$
QUAD-1A	1.82	1.26	2.26	1.54
QUAD-1B	4.23	3.52	5.05	4.45
QUAD-1C	2.03	0.77	2.42	0.88
QUAD-2A	11.04	7.47	11.10	10.87
QUAD-2B	7.09	2.14	9.83	2.91
QUAD-2C	3.73	1.70	4.95	2.09
QUAD-3A	15.16	7.25	17.57	10.72
QUAD-3B	7.80	6.42	10.11	8.72
QUAD-3C	19.72	4.95	27.74	6.37
QUAD-4A	7.14	6.10	10.88	9.88
QUAD-4B	29.38	11.75	53.77	20.81
QUAD-4C	6.59	5.32	6.43	9.01
Average % improvement	0.00%	+42.07%	−29.49%	+17.48%

right-hand-side coefficients, the average CPU time actually *increased* because of the greater processing time required per subproblem.

Based on these test problems, we see that the generalized capacity improvement was most effective in reducing storage requirements (as measured by the number of subproblems evaluated) whereas the capacity improvement applied to only the lower and upper bounds was most effective in reducing the computational time.

5. Summary and Further Work

This paper has presented a generalization of the capacity improvement procedure by applying the domain reduction technique to the right-hand-side coefficients of the constraint set as well as to the simple lower and upper variable bounds. When applied to a set of fixed-charge and concave quadratic test problems, the generalized capacity improvement procedure dramatically reduced the number of subproblems that had to be evaluated in a branch-and-bound algorithm. The generalized capacity improvement procedure also reduced the average amount of CPU time required to solve the test problems. However, an even greater reduction in CPU time was achieved by applying the capacity improvement procedure only to the lower and upper bounds. This was because when the lower and upper bounds for a subproblem were improved, a correspondingly tighter lower envelope to the objective function of the subproblem was obtained. This tighter lower envelope, in turn, produced a tighter relaxation to the subproblems, thus enabling the subproblems to be fathomed more efficiently in the branch-and-bound procedure.

Our future work will focus on two areas. First, penalty methods, which have been used with integer programming branch-and-bound algorithms, can also be applied to concave minimization problems [3]. 'Up and down' penalties have been combined with capacity improvement [16] and we are currently looking at methods for combining capacity improvement with other types of penalties. Second, we have also been exploring the use of nonlinear lower envelopes to produce tighter relaxations to subproblems in the branch-and-bound procedure [1]. The nonlinear envelopes can be combined with penalties and capacity improvement procedures to further increase the efficiency of a branch-and-bound algorithm used to solve nonconvex optimization problems.

Acknowledgements

This work was supported, in part, by research grants from the Energy Modelling Research Group, University of Canterbury, Christchurch, New Zealand and from the Institute of Transportation Studies, University of California, Irvine. Their support is gratefully acknowledged. I also wish to thank Gavin Bell, Mary Anne Lamar, and the referees for their thoughtful reading of the manuscript.

Appendix

This appendix describes how the rate of increase α_i (used in (6) in Section 3.1) can be obtained. A similar approach can be used to determine the value of β_i, γ_k , and δ_k (used in eqs. (10), (14), and (18), respectively). As in Section 3, we assume that we are focusing on the rth iteration of subproblem Q. Thus, in this appendix, we omit the subscript 'Q' and the superscript 'r'.

We assume that the solution to problem $\overline{Q}$ is available for the rth iteration. To describe this solution, we let $J = \{1, 2, \ldots, n\}$ and we partition J into $J = J^B \cup J^L \cup J^U$ corresponding to the basic, nonbasic (at lower bound), and nonbasic (at upper bound) variables. We also partition $\underline{x} = (\underline{x}^B, \underline{x}^L, \underline{x}^U)$, $\underline{c} = (\underline{c}^B, \underline{c}^L, \underline{c}^U)$, and $\underline{A} = (\underline{B}, \underline{A}^L, \underline{A}^U)$. We assume that $\underline{B}$ is invertible. Let z denote the basic variable associated with the objective function and let f be a scalar constant such that $\overline{\phi}(\underline{x}) = f + \underline{c}\underline{x}$. Using this notation, the initial simplex tableau for problem $\overline{Q}$ is given by

$$\begin{array}{c|ccc|c} -z & \underline{x}^B & \underline{x}^L & \underline{x}^U & \text{RHS} \\ \hline 1 & \underline{c}^B & \underline{c}^L & \underline{c}^U & -f \\ \underline{0} & \underline{B} & \underline{A}^L & \underline{A}^U & \underline{b} \end{array} \tag{20}$$

Letting $\overline{\underline{A}}^L = \underline{B}^{-1}\underline{A}^L$, $\overline{\underline{A}}^U = \underline{B}^{-1}\underline{A}^U$, $\overline{\underline{b}} = \underline{B}^{-1}\underline{b}$, $\overline{\underline{c}}^L = \underline{c}^L - \underline{c}^B\overline{\underline{A}}^L$, $\overline{\underline{c}}^U = \underline{c}^U - \underline{c}^B\overline{\underline{A}}^U$, and $\overline{z} = -f - \underline{c}^B\overline{\underline{b}}$, the final simplex tableau for problem $\overline{Q}$ is given by

$$\begin{array}{c|ccc|c} -z & \underline{x}^B & \underline{x}^L & \underline{x}^U & \text{RHS} \\ \hline 1 & \underline{0} & \overline{\underline{c}}^L & \overline{\underline{c}}^U & \overline{z} \\ \underline{0} & \underline{I} & \overline{\underline{A}}^L & \overline{\underline{A}}^U & \overline{\underline{b}} \end{array} \tag{21}$$

where $\underline{I}$ is the identity matrix. Now let $\underline{d}_i$ in constraint (4) be partitioned into $\underline{d}_i = (\underline{d}_i^B, \underline{d}_i^L, \underline{d}_i^U)$, and let s_i denote the slack variable associated with constraint (4). In addition, let $\overline{\underline{d}}_i^L = \underline{d}_i^L - \underline{d}_i^B\overline{\underline{A}}^L$, $\overline{\underline{d}}_i^U = \underline{d}_i^U - \underline{d}_i^B\overline{\underline{A}}^U$, and $\overline{\theta} = \theta - \underline{d}_i^B\overline{\underline{b}}$. Then, the current simplex tableau for the augmented problem $\overline{Q} \mid \underline{d}_i\underline{x} \le \theta$ is given by

$$\begin{array}{c|cccc|c} -z & \underline{x}^B & \underline{x}^L & \underline{x}^U & s_i & \text{RHS} \\ \hline 1 & \underline{0} & \overline{\underline{c}}^L & \overline{\underline{c}}^U & 0 & \overline{z} \\ \underline{0} & \underline{I} & \overline{\underline{A}}^L & \overline{\underline{A}}^U & \underline{0} & \overline{\underline{b}} \\ 0 & \underline{0} & \overline{\underline{d}}_i^L & \overline{\underline{d}}_i^U & 1 & \overline{\theta} \end{array} \tag{22}$$

If $\theta < b_i + \overline{x}_{k(i)}$, then tableau (22) is dual feasible but not primal feasible and we set α_i equal to the rate of increase of the objective function following one iteration of the dual simplex algorithm.

To compute the value of α_i, we consider three cases. First, if $k(i) \in J^L$, then $\overline{\underline{d}}_i = (\underline{0}, \overline{\underline{d}}_i^L, \overline{\underline{d}}_i^U) = \underline{e}_{k(i)}$, (where $\underline{e}_{k(i)} \in \mathbf{R}^n$ is a unit row vector). Since $k(i) \in J^L$, this means that the dual is unbounded, the primal is infeasible, and $\alpha_i = +\infty$. Second, if $k(i) \in J^U$, then once again $\overline{\underline{d}}_i = \underline{e}_{k(i)}$. But now, since $k(i) \in J^U$, this means that $x_{k(i)}$ is the unique variable to enter the basis. So, $\alpha_i = -\overline{c}_{k(i)}$ (where $\overline{c}_{k(i)}$ is the $k(i)$th element of the reduced cost vector $\overline{\underline{c}} = (\underline{0}, \overline{\underline{c}}^L, \overline{\underline{c}}^U)$ in tableau (21)). Finally, if $k(i) \in J^B$, then $\overline{\underline{d}}_i = \overline{\underline{a}}_i$ (where $\overline{\underline{a}}_i$ is the ith row of the reduced constraint matrix $\overline{\underline{A}} = (\underline{I}, \overline{\underline{A}}^L, \overline{\underline{A}}^U)$ in tableau (21)). In this case we define λ_i^L and λ_i^U as

$$\lambda_i^L = \min_{j \in J^L} \left\{ \frac{-\overline{c}_j}{\overline{a}_{ij}} : \overline{a}_{ij} < 0 \right\}, \tag{23}$$

$$\lambda_i^U = \min_{j \in J^U} \left\{ \frac{-\overline{c}_j}{\overline{a}_{ij}} : \overline{a}_{ij} > 0 \right\}. \tag{24}$$

Then the variable entering the basis will be the nonbasic variable associated with the minimum value of λ_i^L and λ_i^U.

In summary, the value of α_i used in (6) in Section 3.1 is computed as follows:

$$\alpha_i = \begin{cases} \min\{\lambda_i^L, \lambda_i^U\} & \text{if } k(i) \in J^B \\ +\infty & \text{if } k(i) \in J^L \\ -\overline{c}_{k(i)} & \text{if } k(i) \in J^U \end{cases} \tag{25}$$

References

1. G. J. Bell, B. W. Lamar, and C. A. Wallace (1995), Capacity Improvement, Penalties, and the Fixed Charge Transportation Problem, Working Paper, Energy Modelling Research Group, Department of Management, University of Canterbury, Christchurch, New Zealand.
2. H. P. Benson (1985), A Finite Algorithm for Concave Minimization over a Polyhedron, *Naval Research Logistics Quarterly* **32**, 165–177.
3. K. M. Bretthauser (1994), A Penalty for Concave Minimization Derived from the Tuy Cutting Plane, *Naval Research Logistics* **41**, 455–463.
4. K. M. Bretthauser, and A. V. Cabot (1994), A Composite Branch and Bound, Cutting Plane Algorithm for Concave Minimization over a Polytope, *Computers and Operations Research* **21**, 777–785.
5. A. V. Cabot (1974), Variations on a Cutting Plane Method for Solving Concave Minimization Problems with Linear Constraints, *Naval Research Logistics Quarterly* **21**, 264–274.
6. J. E. Falk, and K. L. Hoffman (1986), Concave Minimization via Collapsing Polytopes, *Operations Research* **34**, 919–929.
7. A. M. Geoffrion, and R. E. Marsten (1972), Integer Programming Algorithms: A Framework and State-of-the-Art Survey, *Management Science* **18**, 465–491.
8. P. Gray (1967), Mixed Integer Programming Algorithm for Site Selection and Other Fixed Charge Problems, Technical Report No. 6, Department of Operations Research, Stanford University, Stanford, CA.
9. C. D. Heising-Goodman (1981), A Survey of Methodology for the Global Minimization of Concave Functions Subject to Convex Constraints, *OMEGA* **9**, 313–319.
10. K. L. Hoffman (1981), A Method for Globally Minimizing Concave Functions Over Convex Sets, *Mathematical Programming* **20**, 22–32.
11. K. L. Hoffman, and M. Padberg (1985), LP-Based Combinatorial Problem Solving, *Annals of Operations Research* **4**, 145–194.
12. R. Horst (1986), A General Class of Branch-and-Bound Methods in Global Optimization with Some New Approaches for Concave Minimization, *Journal of Optimization Theory and Applications* **51**, 271–291.
13. R. Horst, and H. Tuy (1993), *Global Optimization: Deterministic Approaches*, 2nd ed. Springer-Verlag, Berlin.
14. B. M. Khumawala (1972), An Efficient Branch and Bound Algorithm for the Warehouse Location Problem, *Management Science* **18**, 718–731.
15. B. W. Lamar (1985), Network Design Algorithms with Applications to Freight Transportation, Ph.D. Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.
16. B. W. Lamar (1993), An Improved Branch and Bound Algorithm for Minimum Concave Cost Network Flow Problems, *Journal of Global Optimization* **3**, 261–287.
17. B. W. Lamar (1993), Nonconvex Optimization Over a Polytope Using Generalized Capacity Improvement, presented at the ORSA/TIMS Joint National Meeting, Phoenix, AZ, November.
18. B. W. Lamar, Y. Sheffi, and W. B. Powell (1990), A Capacity Improvement Lower Bound for Fixed Charge Network Design Problems, *Operations Research* **38**, 704–710.
19. G. P. McCormick (1972), Attempts to Calculate Global Solution of Problems that May Have Local Minima, in F. A. Lootsma (Ed.), *Numerical Methods for Nonlinear Optimization*, 201–221, Academic Press, New York.
20. P. M. Pardalos (1994), On the Passage from Local to Global in Optimization, in J. R. Birge and K. G. Murty (Eds.), *Mathematical Programming: State of the Art 1994*, 220–247, The University of Michigan Press, Ann Arbor, MI.
21. P. M. Pardalos, and J. B. Rosen (1986), Methods for Global Concave Minimization: A Bibliographic Survey, *SIAM Review* **28**, 367–379.
22. P. M. Pardalos, and J. B. Rosen (1987), *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science **268**, Springer-Verlag, Berlin.
23. H. S. Ryoo, and N. V. Sahinidis (1993), Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design, Technical Report UILU-ENG 93-4023, Department of

Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, November.
24. J. P. Shectman, and N. V. Sahinidis (1995), A Finite Algorithm for Global Minimization of Separable Concave Programs, Working Paper, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, January.
25. N. V. Thakur (1990), Domain Contraction in Nonlinear Programming: Minimizing a Quadratic Concave Function over a Polyhedron, *Mathematics of Operations Research* **16**, 390–407.
26. H. Tuy (1964), Concave Programming under Linear Constraints, *Soviet Mathematics* **5**, 1437–1440.