A decomposition method for quadratic programming

by D. L. Jensen
A. J. King

We discuss the algorithms used in the Optimization Subroutine Library for the solution of convex quadratic programming problems. The basic simplex algorithm for convex quadratic programming is described. We then show how the simplex method for linear programming can be used in a decomposition crash procedure to obtain a good initial basic solution for the quadratic programming algorithm. We show how this solution may be used as a starting solution for the simplex-based algorithm.

Besides its ability to obtain good starting solutions, this procedure has several additional properties. It can be used directly to find an optimal solution to a quadratic program instead of simply finding a good initial solution; it provides both upper and lower bounds on the objective function value as the algorithm proceeds; it reduces the complexity of intermediate calculations; it avoids certain numerical difficulties that arise in quadratic, but not linear programming.

In this paper we review the basic algorithms for convex quadratic programming that are part of the Optimization Subroutine Library. The quadratic programs have linear constraints, the variables are subject to nonnegativity constraints, and the objective function has a linear and a quadratic part where the quadratic part is convex.

One of the motivating applications for quadratic programming is the Markowitz[1] model of risk that is used in portfolio analysis. In this application, the linear term of the objective function is used to measure the expected return of a portfolio, while the quadratic term is used to measure the risk or variation about the mean associated with the portfolio. It provides investors with a method to balance return and risk in selecting a portfolio of investments. Other applications include asymmetric risk models for portfolio analysis, least-squares problems, proximal point algorithms, and sequential quadratic programming for nonlinear programming.

One property of quadratic programs that makes them potentially more difficult to solve than linear programs with the same number of variables and constraints is that, unlike the solution to a linear program, the solution to a quadratic program may necessarily use all the variables of the problem. In the application of portfolio analysis, this reflects the practice of hedging against risk by diversifying. In the context of solving quadratic programs, this property is manifested in the difficulty of finding good initial feasible solutions to a quadratic program.

In this paper, we begin by discussing the optimality conditions for a quadratic program. We use these conditions to motivate the development of an algorithm for quadratic programming based on the simplex method for linear programming.

This algorithm is essentially that of Dantzig.[2] Given such an algorithm, we discuss a decomposition procedure that uses our quadratic programming algorithm, on a related but much smaller quadratic program, in conjunction with the simplex algorithm for linear programming to produce a good starting solution to the original quadratic programming problem. This is the first phase of our solution procedure for quadratic programming problems. The procedure used in this is a decomposition procedure since it uses a subproblem to generate good feasible solutions to the quadratic program. This decomposition approach is closely related to both Dantzig-Wolfe decomposition[3] and simplicial decomposition.[4]

We describe a procedure for converting the starting solution produced by the decomposition procedure into one that can be used by the simplex-based quadratic programming algorithm. The second phase of the algorithm consists of moving from the starting solution output by the first phase to an optimal solution to the quadratic program. We conclude the paper with a brief discussion of computational experience with this two-phase approach to convex quadratic programming.

Simplex-based algorithm for quadratic programming

The problem we have in mind is (QP):[5]

$$
\begin{array}{lll}
(QP) & \text{minimize:} & c^{\mathsf{T}}x + \tfrac{1}{2}x^{\mathsf{T}}Qx \\
 & \text{subject to:} & Ax = b \\
 & & x \geq 0
\end{array}
$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m}$, $c \in \mathbb{R}^{n}$, $x \in \mathbb{R}^{n}$, and $Q \in \mathbb{R}^{n\times n}$ is positive semidefinite. The requirement that Q is positive semidefinite is what makes the quadratic program convex and is equivalent to the requirement that

$$z^{\mathsf{T}}Qz \geq 0$$

for all $z \in \mathbb{R}^{n}$. For simplicity, we assume that the feasible region $\mathcal{P}$ of (QP) is nonempty,

$$\mathcal{P} = \{x \mid Ax = b,\ x \geq 0\} \neq \varnothing \tag{1}$$

and has no nonnegative directions of descent,

$$\{z \mid Qz = 0,\ Az = 0,\ (c + Qx)^{\mathsf{T}}z < 0,\ z \geq 0,\ x \in \mathcal{P}\} = \varnothing \tag{2}$$

The first of these requirements is simply that the program under consideration has some solution. The second is equivalent to the requirement that the program has an optimal solution. A final technical requirement is that A has full row rank.

One approach to solving quadratic programs is to generalize the simplex method for linear programming. In order to describe this method we need to discuss the dual of the problem (QP)

$$
\begin{array}{lll}
(QD) & \text{maximize:} & y^{\mathsf{T}}b - \tfrac{1}{2}u^{\mathsf{T}}Qu \\
 & \text{subject to:} & A^{\mathsf{T}}y - Qu \leq c \\
 & & u \text{ unrestricted} \\
 & & y \text{ unrestricted}
\end{array}
$$

The dual program is useful in two regards. First, the value of each feasible solution to the dual program is lower than the value of each feasible solution to the primal program. This is easy to see. If x is feasible in (QP) and (y, u) is feasible in (QD), then

$$(c - A^{\mathsf{T}}y + Qu)^{\mathsf{T}}x \geq 0$$

which is equivalent to

$$c^{\mathsf{T}}x - y^{\mathsf{T}}b + x^{\mathsf{T}}Qu \geq 0$$

Since Q is positive semidefinite we have that

$$
\begin{aligned}
0 &\leq \tfrac{1}{2}(x - u)^{\mathsf{T}}Q(x - u) \\
&= \tfrac{1}{2}x^{\mathsf{T}}Qx - x^{\mathsf{T}}Qu + \tfrac{1}{2}u^{\mathsf{T}}Qu
\end{aligned}
$$

and combining this with the last inequality above gives the stated relation between primal and dual objective function values

$$c^{\mathsf{T}}x + \tfrac{1}{2}x^{\mathsf{T}}Qx \geq y^{\mathsf{T}}b - \tfrac{1}{2}u^{\mathsf{T}}Qu \tag{3}$$

This relation is the *weak duality* relationship of the dual pair of quadratic programs (QP) and (QD). The second property of (QP) and (QD) is that if both (QP) and (QD) have optimal solutions, then there is a solution pair of the same value. Furthermore, if x is optimal for (QP), then there exists y, so that (y, x) is optimal for (QD). This property is called the *strong duality* property. Due to our assumptions of feasibility and boundedness, both programs have optimal solutions.

In order to argue that the strong duality property holds, we consider the following *augmentation subproblem* $(AP(x))$ defined with respect to a feasible solution x to the primal problem (QP)

$$(AP(x)) \qquad (c + Qx)^{\mathsf{T}} z < 0$$

$$\text{where:} \quad \begin{array}{ll} Az = 0 & \\ z_j \geq 0 & \text{whenever } x_j = 0 \\ z_j \ \text{unrestricted} & \text{whenever } x_j > 0 \end{array}$$

We call this the augmentation subproblem because whenever it has a solution z, that solution can be used to augment x to a solution $x + \theta z$ of strictly lower value. Suppose $AP(x)$ has no solution. We argue that this implies that x is optimal. If $AP(x)$ has no solution, then an application of Farkas's Lemma allows us to conclude that the linear system

$$\begin{array}{ll} (A^{\mathsf{T}}y)_j \leq (c + Qx)_j & \text{whenever } x_j = 0 \\ (A^{\mathsf{T}}y)_j = (c + Qx)_j & \text{whenever } x_j > 0 \end{array} \qquad (4)$$

has a solution. It is easy to see that if the relations in Equation 4 hold, then the weak duality inequality (Equation 3) holds with equality. This implies that x has the same value in (QP) as (y, x) has in (QD), and it is also easy to verify that (y, x) is feasible for (QD).

The relations in Equation 4 are stronger than the relations usually referred to as the complementarity relations, which state that for $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $u \in \mathbb{R}^n$, then x and (y, u) are *complementary* if $x_j(c - A^{\mathsf{T}}y + Qu)_j = 0$ for $1 \leq j \leq n$. Simplex-based algorithms for quadratic programming fall into the same framework as simplex-based algorithms for linear programming. Of the three properties

1. Primal feasibility
2. Complementarity
3. Dual feasibility

a primal simplex algorithm maintains the first two, while a dual simplex algorithm maintains the last two. Simplex algorithms do this by imposing the further restriction of moving from one basic solution to another using pivots.

Restating the observations outlined above, we know that we can solve a convex quadratic program if we can find a solution to the system of equations

$$Ax = b$$

$$A^{\mathsf{T}}y - Qx + s = c$$

with

$$y \text{ unrestricted} \qquad x \geq 0,\ s \geq 0$$

that satisfies the additional complementarity condition

$$s_j x_j = 0 \qquad \text{for } 1 \leq j \leq n \qquad (5)$$

In matrix terms this system of equations can be expressed as

$$\begin{bmatrix} 0 & A & 0 \\ A^{\mathsf{T}} & -Q & I \end{bmatrix} \begin{bmatrix} y \\ x \\ s \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} \qquad (6)$$

In order to proceed with the description of the simplex method for convex quadratic programming, some new terms are introduced for clarity.

A set of variables indexing a full rank submatrix of the matrix in Equation 6 is called a *basis B* for the quadratic programming problem. We say a solution (y, x, s) *uses* a variable if it is nonzero on that variable. A solution (y, x, s) to the system of equations above is called *basic* if it is the unique solution to this system of equations using only the variables in a basis B. Given a basic solution (y, x, s), we call the solution *primal feasible* if x is nonnegative, *dual feasible* if s is nonnegative, and *complementary* if Equation 5 holds. We call a basis *complementary B* if all the variables y are in the basis and $x_j \in B$ if and only if $s_j \notin B$. Note that if a basis B is complementary, then the associated basic feasible solution (y, x, s) satisfies the condition in Equation 5.

If we are using a primal simplex method, we will move from one primal feasible complementary basic solution to another. This will be done by alternately performing two operations called *pricing* and *pivoting*. The pricing operation determines a variable x_j to introduce into the basis. The pivoting operation determines a new complementary basis containing the variable x_j. We now discuss these two operations.

Pricing. Without worrying about how we obtain an initial solution, suppose that we have a primal feasible basic solution (y, x, s) with respect to a complementary basis B. We may rewrite the optimality conditions described in Equation 4 as

$$s_j \geq 0 \qquad \text{whenever } x_j = 0$$

$$s_j = 0 \qquad \text{whenever } x_j > 0$$

Since (y, x, s) is complementary, checking this condition is equivalent to checking whether s is nonnegative. This is called the *pricing* operation of the simplex method. If some variable s_j is less than 0, then it is beneficial to introduce the variable x_j into the basis B. In particular, there is a unique vector (w, z, t) with $z_j = +1$ and

$$\begin{bmatrix} 0 & A & 0 \\ A^{\mathsf{T}} & -Q & I \end{bmatrix} \begin{bmatrix} w \\ z \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which only uses variables in $B \cup \{x_j\}$. Since B is a complementary basis, the inner product of s and z is equal to s_j. But this number s_j is also equal to

$$(c - A^{\mathsf{T}}y + Qx)^{\mathsf{T}}z = (c + Qx)^{\mathsf{T}}z$$

That is to say, the negative number s_j is equal to the directional derivative of the objective function at the point x in the direction z. Thus, for at least some small step in the direction z, the objective function decreases. So the pricing operation either allows us to conclude that x is an optimal solution to (QP), in the case where s is nonnegative, or provides a direction z of potential objective function improvement.

Pivoting. Once we have an augmenting direction z, we use that direction to update our current solution x. We must determine the maximum length step we can take in this direction while preserving primal feasibility. Additionally, after taking such a step, we must convert the new and improved solution to a complementary basic feasible solution. We refer to this operation as *pivoting*. While in the case of linear programming this step from one basic feasible complementary solution to another requires a single basis exchange, in the case of quadratic programming this may require several basis exchanges.

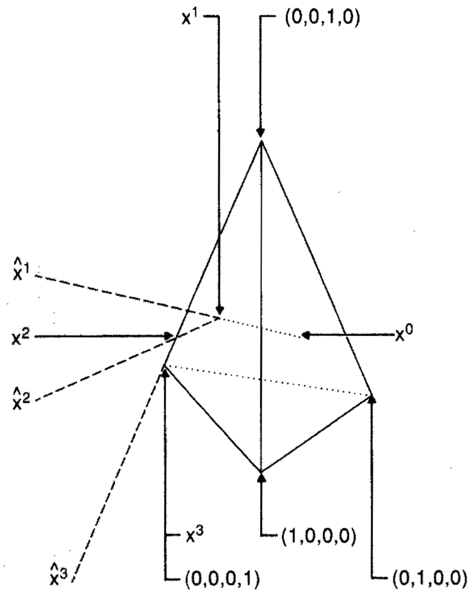
The procedure that we describe is illustrated in Figure 1. In this example, suppose the solution at the beginning of a pivot sequence is x^0. It is the optimal solution to the quadratic programming problem when variable x_4 is not used. If, however, we introduce x_4 into this solution, while maintaining complementarity on variables x_1, x_2, and x_3, this defines a unique direction passing through the points x^0, x^1, and $\hat{x}^1$ of the figure. The point $\hat{x}^1$ optimizes the quadratic function along this line segment, but violates the nonnegativity requirements on the first three coordinates. The best feasible point on this line segment is x^1, so we move to x^1. Note that the complementarity condition is violated on the pair of variables x_4 and s_4. This argument is now repeated. We maintain the conditions $x_1 = 0$ and complementarity on the variables x_2 and x_3. These conditions define a unique direction passing through the points x^1, x^2, and $\hat{x}^2$ of the figure. The best feasible point on this line segment is x^2, which violates nonnegativity on coordinates 2 and 3. This argument is repeated once more. The point x^3 represents a basic feasible complementary solution.

We now describe this operation in greater detail. Given a direction z, the pivoting operation then pursues this direction in order to find a new complementary basis of improved value. We consider solutions of Equation 6 along the half line $(y(\theta), x(\theta), s(\theta)) = (y, x, s) + \theta(w, z, t)$ where θ is greater than 0.

Since z_j is greater than 0, x_j is equal to 0, and s_j is less than 0, solutions on this half line are not complementary except for possibly one value of θ. To define this value of theta, we first observe that if $t_j \neq 0$, then $t_j > 0$ by the convexity of the objective function. To see why $t_j \geq 0$, consider the product

$$[w \quad z]\begin{bmatrix} 0 & A \\ A^{\mathsf{T}} & -Q \end{bmatrix}\begin{bmatrix} w \\ z \end{bmatrix} eq - z^{\mathsf{T}}Qz$$

Figure 1 Pivoting in simplex algorithms for quadratic programming



$$\min -4/3x_1 - 8/3x_2 - 4x_3 - 10x_4 + 2x_1^2 + 4x_2^2 + 6x_3^2 + \tfrac{1}{2}x_4^2$$
$$x_1 + x_2 + x_3 + x_4 = 1,\ x \geq 0$$

	x_1	x_2	x_3	x_4	VALUE
x^n	1/3	1/3	1/3	0	−1 1/3
s^n	0	0	0	−10	
x^1	0	3/18	4/18	11/18	−6 551/648
s^1	0	0	0	−145/18	
$\hat{x}^1$	−29/21	−11/2	−5/21	66/21	−17 1/21
x^2	0	0	1/9	8/9	−8 70/81
s^2	4/3	0	0	−58/9	
$\hat{x}^2$	0	−2/3	−1/3	2	−12 4/9
x^3	0	0	0	1	− 9 1/2
s^3	23/3	19/3	5	0	
$\hat{x}^3$	0	0	−5/13	18/13	−10 6/13

which is nonpositive since Q is positive semidefinite. But the definition of (w, z, t) implies that this product is also equal to $-z^{\mathrm{T}}t$. The complementarity of the basis B implies $z^{\mathrm{T}}t = t_j$ and so $t_j = z^{\mathrm{T}}Qz$. Given $t_j > 0$, $(y(\theta), x(\theta), s(\theta))$ is complementary for the single value $\theta^* = -(s_j/t_j)$. If $(y(\theta^*), x(\theta^*), s(\theta^*))$ is also primal feasible, then the new basis is $B\backslash\{s_j\} \cup \{x_j\}$. (In the example of Figure 1 this case never occurs.)

Otherwise, either $t_j = 0$ or there is some variable x_k such that $x_k + \theta^* z_k$ is less than zero. In the latter case we cannot move as far along the half line as $(y, x, s) + \theta(w, z, t)$ while remaining primal feasible. In either case, the scalar θ is redefined using a minimum ratio test

$$\theta = \min\left\{-\frac{x_k}{z_k}\middle| z_k < 0\right\}$$

so as to maximize the steplength in the direction z while preserving primal feasibility. We know that there is at least one index k with $z_k < 0$ since $\mathcal{P}$ is bounded. If x_k is a variable such that $\theta = -(x_k/z_k)$, then $(y(\theta), x(\theta), s(\theta))$ is a primal feasible basic solution with respect to the basis $B\backslash\{x_k\} \cup \{x_j\}$. Although this basis is not complementary, the convexity of the objective function ensures $B^1 = B\backslash\{x_k\} \cup \{s_k\}$ is a complementary basis and the associated basic solution $(\hat{y}^1, \hat{x}^1, \hat{s}^1)$ has a better value than $(y(\theta), x(\theta), s(\theta))$. It is in fact the solution to the quadratic program where the variables x_j in B^1 are set unrestricted in sign while the variables x not in B^1 are fixed at zero. If the solution $(\hat{y}^1, \hat{x}^1, \hat{s}^1)$ is primal feasible, then we take that point as our next complementary basic feasible solution.

Otherwise, consider iteratively applying the following argument. We suppose we have a primal feasible basic solution (y^k, x^k, s^k) and a complementary basis B^k containing all the variables x used by x^k. If the basic solution say $(\hat{y}^k, \hat{x}^k, \hat{s}^k)$ associated with B^k is feasible, then it becomes our new primal feasible complementary solution. Otherwise, θ defined by

$$\theta = \min \left\{ \frac{x_j}{x_j - \hat{x}_j} \middle| \hat{x}_j < 0 \right\}$$

is positive. Furthermore, the set B^{k+1} indexed by $B^k \backslash \{x_j\} \cup \{s_j\}$ is a complementary basis where j is a binding index in determining θ. This fact is not immediately obvious, but follows from the assumption that Q is positive semidefinite. The argument is now repeated with B^{k+1} replacing B^k. Note that at each pass of the argument presented in this paragraph, the number of variables x_j in the basis B^k decreases by one. Once this number of variables reaches m, the basis is guaranteed to be a complementary primal feasible basis.

Decomposition for quadratic programming

We have previously explained how the simplex method can be generalized in a fairly natural way to derive algorithms for convex quadratic programming. However, there are certain difficulties that arise in quadratic programming that do not arise in linear programming that make the practicality of such an algorithm questionable. We first note that although all bases have cardinality $m + n$, the effective size of a basis is the m plus the number of variables x_j in the basis. This is because the variables s_j index unit columns and their values can be easily determined once the values of y and x are set. The number of x variables in the basis is bounded above by the rank of A plus the rank of Q. In the case of linear programming, the basis always has m variables.

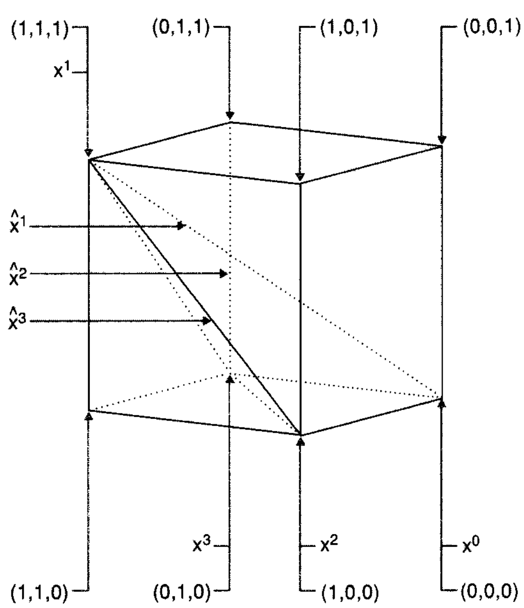
A second problem that arises in practice is that the matrices Q and A may be badly scaled relative to each other. One effective technique for handling badly scaled problems in linear programming applications is to scale the constraint matrix. In the case of quadratic programming, scaling applied to the constraint matrix must also be applied to the quadratic part of the objective function Q. Since Q and A may be out of scale with respect to each other, this technique may not be as effective.

In order to solve convex quadratic programming problems and overcome these difficulties, we want to capitalize on the relative simplicity of solving linear programming problems. The toolbox we have at hand is the Optimization Subroutine Library (OSL). In OSL there is a fast and reliable simplex routine for linear programming, tuned to take advantage of the various hardware platforms on which OSL runs. OSL also has a simplex-based approach for quadratic programming based on the ideas sketched in the section describing the standard form convex quadratic program. The library also provides a framework by which different mathematical programs can communicate information to each other.

The algorithm described in this section represents an attempt to exploit these attributes of the Optimization Subroutine Library while overcoming some of the problems mentioned at the beginning of the section. In order to motivate the procedure, we reconsider the primal quadratic program (QP) and the augmentation subproblem $(AP(x))$. Given a feasible solution x^0 to (QP) we recall that determining whether x^0 is optimal is equivalent to determining whether $AP(x^0)$ has a feasible solution. But $AP(x^0)$ is a linear system and so this can easily be tested using linear programming. Thus, an algorithm for solving quadratic programs could consist of alternately solving $AP(x^0)$ to find an augmenting direction z^0 and then using z^0 to update x^0 to a new and better solution to the quadratic program x^1.

This approach, however, is bound to converge very slowly in most circumstances. A natural modification is to use information not only from the current solution of $AP(x^k)$ but also from previous iterations. This idea is illustrated in Figure 2. In this example, we suppose that x^0 is our initial solution. We find that the direction leading from the point x^0 to the point x^1 is an improving direction. The point along that direction which optimizes the quadratic objective function is labeled $\hat{x}^1$. From this point we see that the direction leading to x^2 is improving. However, instead of optimizing the quadratic objective over the line segment joining $\hat{x}^1$ and x^2 we optimize over the triangle (simplex) with corners x^0, x^1, and x^2. The optimal solution is given by $\hat{x}^2$. Again we look for an improving direction from out of the current point and find moving toward x^3 improves the quadratic objective function. We now optimize over the simplex with corners x^0, x^1, x^2, and x^3. The optimal solution to this subproblem is optimal for the entire problem. The method of collecting the points x^k is now formalized.

Figure 2 Decomposition algorithm for quadratic programming



$$\min -15x_1 - 12x_2 - 9x_3$$
$$+ 9x_1^2 + 9x_2^2 + 9x_3^2$$
$$0 \le x \le 1$$

	x_1	x_2	x_3	VALUE	BOUND
x^n	0	0	0	0	
x^1	1	1	1	−9	−36
$\hat{x}^1$	2/3	2/3	2/3	−12	
x^2	1	0	0	−6	−15
$\hat{x}^1$	5/6	7/12	7/12	−12 3/8	
x^3	0	1	0	−3	−13 7/8
$\hat{x}^3$	5/6	2/3	1/2	−12 1/2	
x^4	5/6	2/3	1/2	−12 1/2	−12 1/2

Suppose we collect the solutions x^k as the columns of a matrix X^k and solve problem $QMP(X^k)$ where QMP is defined by

$$(QMP(X)) \qquad \text{minimize:} \qquad c^{\mathsf{T}}X\lambda + \tfrac{1}{2}\lambda^{\mathsf{T}}X^{\mathsf{T}}QX\lambda$$
$$e^{\mathsf{T}}\lambda = 1$$
$$\lambda \ge 0$$

Since each of the solutions x^i comprising the columns of X^k is a feasible solution to QP, the constraints

$$AX\lambda = b$$

$$X\lambda \ge 0$$

are satisfied implicitly. Any feasible solution λ to $QMP(X)$ corresponds to a feasible solution $X\lambda$ to QP since the matrix X has columns that are feasible for QP.

Although QMP is a quadratic programming problem, it has a single linear constraint. The rank of the matrix $X^{\mathsf{T}}QX$ in the objective function is bounded above by the number of columns collected in the matrix X. Furthermore, the solution to $QMP(X)$ is guaranteed to have lower value than the value of any of the columns of X.

$AP(x)$ can be used to collect points x^i to add to X^k. After enough points are collected, an optimal solution to QP will be obtained. However, one of the drawbacks of this approach is a problem that is also present in the method described in the section on the simplex-based algorithm for quadratic programming. Namely, neither method provides lower bounds on the value of the problem (QP) as it proceeds.

This difficulty can be overcome if we solve a potentially more difficult problem than $AP(x)$ as a subproblem of the overall procedure. This subproblem is similar to the problem $AP(x)$ and is defined as $QSP(x)$.

$$(QSP(x)) \qquad \text{minimize:} \qquad \begin{aligned}&(c + Qx)^{\mathsf{T}}z\\ &Az = 0\\ &z \geq -x\end{aligned}$$

This problem always has an optimal solution when $\mathcal{P}$ satisfies the feasibility assumptions in Equations 1 and 2. We claim that the optimal solution of this subproblem provides a lower bound on the value of the program (QP). In order to see this we consider the dual of this program $QSD(x)$.

$$(QSD(x)) \quad \text{minimize:} \quad \begin{aligned}&-\gamma^{\mathsf{T}}x\\ &A^{\mathsf{T}}y + \gamma = c + Qx\\ &y \text{ unrestricted} \qquad \gamma \geq 0\end{aligned}$$

We first observe that, due to the nonnegativity of γ, any feasible solution (y, γ) to $QSD(x)$ may be combined with x to form a feasible solution (y, x) to QD. Furthermore, the value of this solution to $QSD(x)$ is the duality gap between the dual feasible solution (y, x) and the primal feasible solution x

$$\begin{aligned}-\gamma^{\mathsf{T}}x &= (A^{\mathsf{T}}y - c - Qx)^{\mathsf{T}}x\\ &= (y^{\mathsf{T}}b - \tfrac{1}{2}x^{\mathsf{T}}Qx) - (c^{\mathsf{T}}x + \tfrac{1}{2}x^{\mathsf{T}}Qx)\end{aligned}$$

If this duality gap is nonzero and z solves $QSP(x)$, then $x + z$ is a better feasible solution to QP than x and should be added to the collection of points X. If the gap is zero then no augmenting direction exists and the current solution x is optimal.

The algorithm suggested by this discussion is given in Figure 3.

In the example, the points of Figure 2 labeled x^1, x^2, and x^3 are the solutions to the problems $QSP(x^0)$, $QSP(\hat{x}^1)$, and $QSP(\hat{x}^2)$, respectively. In addition, the solutions of the subproblems give lower bounds on the objective function values, which are also listed in Figure 2.

Figure 3 The decomposition algorithm

1. Initialization

Solve the linear program obtained from dropping the matrix Q from (QP) to obtain a feasible point x^0.

$X^0 \leftarrow [x^0]$; $k \leftarrow 0$;

2. Master problem step

Solve $QMP(X^k)$ to obtain a solution $\hat{x}^k = X^k\lambda^*$.

3. Subproblem step

Solve $QSP(\hat{x}^k)$ to obtain a solution x^{k+1}.

a. Test optimality

If $(c + Q\hat{x}^k)^{\mathsf{T}}x^{k+1} = 0$

then **stop** with an optimal solution $\hat{x}^k$.

b. Continue by augmenting X^k.

$X^{k+1} \leftarrow [X^k \mid \hat{x}^k + x^{k+1}]$;

$k \leftarrow k + 1$;

Go to 2;

Termination

It can be argued that the decomposition algorithms discussed are finite algorithms. In practice, however, it is not advisable to run these algorithms to completion, but to instead use them to find a good starting solution for the algorithm. In order to implement the starting solution, one needs to be able to convert a solution of $QMP(X^k)$ to a basic feasible complementary solution to QP. This can be accomplished using ideas outlined earlier during the discussion of pivoting.

We suppose that we have a complementary basis B^0 and a primal feasible solution x^0. In order to convert x^0 to a basic feasible complementary solution requires two passes of the variables. In the first pass, we convert x^0 to a basic feasible solution by choosing a variable x_j where x_j is not basic, but has $x_j^0 > 0$. Then we consider the unique direction z^0 satisfying $z_j^0 = +1$ and

$$\begin{bmatrix} 0 & A & 0 \\ A^{\mathsf{T}} & -Q & I \end{bmatrix} \begin{bmatrix} w \\ z \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which only uses variables in $B^0 \cup \{x_j\}$. Either z^0 or $-z^0$ is augmenting with respect to x. Call this direction $\bar{z}^0$. We then take a step in that direction of length θ where θ is the maximum distance that we can move while decreasing the objective function value and maintaining the feasibility of $x^1 = x^0 + \theta\bar{z}^0$. If the minimizer of the quadratic function along this half line occurs before any variable x_j goes to zero, then the quadratic function must have curvature along this half line and so $t_j^0 \neq 0$. This then implies that $B^1 = B_0\backslash\{s_j\} \cup \{x_j\}$ is a complementary basis. Otherwise, some variable component of x^1 corresponding to a variable x_k with $\bar{z}_k^0 < 0$ goes to zero. It turns out that $B^1 = B^0\backslash\{x_k, s_j\} \cup \{x_j, s_k)$ is a complementary basis. In either of the two cases above, the number of variables that x^1 uses and that are not in B^1 is at least one less than the number of variables that x^0 uses and that are not in B^0. Additionally, either x^1 does not use x_j or $x_j \in B^1$. Thus, in one pass of the variables we can obtain a feasible solution x^k and a complementary basis B^k so that every variable that x^k uses is also in B^k.

But now we are in the situation already discussed in the last paragraph in the section on pivoting. After one additional pass over the variables we will have a complementary basic feasible solution and thus can initiate the algorithm.

Conclusions

As an example to illustrate the usefulness of the decomposition approach to convex quadratic programming, we will consider a problem from NETLIB,[6] PILOTNOV. The problems in NETLIB are linear programming problems, and so we need to add a positive semidefinite matrix to the data to create a convex quadratic programming problem. The matrix we add is tridiagonal with twos on the diagonal and negative ones off the diagonal. The linear programming problem has 975 rows, 2172 columns, and 13 057 nonzeros, and the positive semidefinite matrix has 6514 additional nonzeros.

If the decomposition algorithm is run for 25 major iterations, an upper bound of 505 882 453 and a lower bound of 501 304 145 are obtained after 6633 linear programming simplex operations. The solution from the decomposition is converted to a basic complementary solution of value 505 112 850 using another 302 pivots. This solution is converted to an optimal solution of value 504 976 497 using 97 pivots of the type outlined in an earlier section. The optimal solution has 1144 basic primal variables and requires the solutions of systems of equations with 2119 variables. On the other hand, each system solved by the simplex algorithm for linear programming uses only 975 variables. The problem is solved in a total of 261 seconds on a RISC System/6000 Model 530 workstation.

If the decomposition approach is limited to five major iterations, the lower bound established has value 490 959 083, the solution value after conversion to a complementary solution has value 505 478 905, and 322 seconds of execution time have been consumed. Only 2656 linear programming pivots have been performed; 176 pivots are required to convert the solution to a complementary basic feasible solution; 219 additional pivots are required to obtain an optimal solution. The total time required is 569 seconds.

The decomposition approach with 25 major iterations allowed us to limit the the number of pivots in this larger system to 399. Each iteration of the quadratic programming code requires more work than an iteration of the linear programming code, and so limiting the number of the expensive iterations is crucial for the efficient solution of quadratic programming problems. If the decomposition approach is not used, the basic simplex algorithm for quadratic programming implemented in the Optimization Subroutine Library is an order of magnitude slower when applied to this problem.

This problem can certainly be considered a relatively difficult problem and the running time improvement observed here is not seen on all problems. However, on all but small quadratic programs, the decomposition approach does give smaller execution times than the pure simplex approach. Furthermore, even a few iterations of the approach often lead to very good starting solutions for the quadratic programming solver.

Cited references and note

1. H. M. Markowitz, *Portfolio Selection Efficient Diversification of Investments*, Wiley, New York (1959).
2. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ (1963).
3. G. B. Dantzig and Philip Wolfe, "Decomposition Principle for Linear Programs," *Operations Research* **8**, No. 1, 101–111 (1960).
4. B. Hohenbalken, "A Finite Algorithm to Maximize Certain

Pseudoconcave Functions on Polytopes," *Mathematical Programming* **8**, 189–206 (1975).

5. The notation (*QP*) identifies a quadratic program. The types of programming problems are identified by initials in parentheses.
6. D. M. Gay, "Electronic Mail Distribution of Linear Programming Test Problems," *Mathematical Programming Society COAL Newsletter* (1985).

Accepted for publication October 7, 1991.

David L. Jensen *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598*. Dr. Jensen received his Ph.D. in operations research from Cornell University in 1985. He joined IBM's Thomas J. Watson Research Center in 1987. His interests are in network flows, combinatorial optimization, interior point algorithms, and applications of mathematical programming.

Alan J. King *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598*. Dr. King earned his Ph.D. at the University of Washington, Seattle, in 1986, and joined IBM's Thomas J. Watson Research Center in 1988. His interests are in stochastic programming and its application to decision-making under uncertainty for engineering and economic systems.

Reprint Order No. G321-5459.