Journal of Global Optimization **24:** 349–370, 2002.
© 2002 *Kluwer Academic Publishers. Printed in the Netherlands.*

A Global Optimization Algorithm using Lagrangian Underestimates and the Interval Newton Method

TIM VAN VOORHIS
Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 2019 Black Engineering, Ames, IA 50011, USA (tvanvoor@iastate.edu)

Abstract. Convex relaxations can be used to obtain lower bounds on the optimal objective function value of nonconvex quadratically constrained quadratic programs. However, for some problems, significantly better bounds can be obtained by minimizing the restricted Lagrangian function for a given estimate of the Lagrange multipliers. The difficulty in utilizing Lagrangian duality within a global optimization context is that the restricted Lagrangian is often nonconvex. Minimizing a convex underestimate of the restricted Lagrangian overcomes this difficulty and facilitates the use of Lagrangian duality within a global optimization framework. A branch-and-bound algorithm is presented that relies on these Lagrangian underestimates to provide lower bounds and on the interval Newton method to facilitate convergence in the neighborhood of the global solution. Computational results show that the algorithm compares favorably to the Reformulation–Linearization Technique for problems with a favorable structure.

Key words: Lagrangian dual, Interval Newton method, Convex underestimate, Quadratically constrained quadratic program

1. Introduction

In this paper, we focus on the quadratically constrained quadratic program with bounded variables:

$$\begin{aligned}(\text{QCQP}) z^* = \text{Min } & f^0(x) \\ \text{subject to } & f^i(x) \leqslant b_i, i = 1, \ldots, m \\ & -\infty < \ell \leqslant x \leqslant u < \infty,\end{aligned}$$

where $f^i(x) = c_i^T x + x^T Q^i x, \ i = 0, 1, \ldots, m$. Without loss of generality, it can be assumed that matrices $Q^i, \ i = 0, \ldots, m$ are symmetric. Quadratically constrained problems are worthy of study both because they frequently appear in applications and because many other nonlinear problems can be transformed into this form.

When each function $f^i(x)$ is convex, then efficient algorithms are available for solving QCQP. For the remainder of this paper, we assume that QCQP is nonconvex. While solving problems of this sort is known to be NP-hard, many practical applications possess a relatively favorable structure that can be exploited. In particular, oftentimes the matrices $Q^i, \ i = 0, \ldots, m$ are relatively sparse.

Examples of sparse nonconvex QCQPs can be found in engineering and operations research literature. Robust control seeks to ensure that controllers will perform adequately even in the presence of uncertainty regarding the current state of the system. The design of robust controllers minimizes a linear or convex quadratic function subject to bilinear constraints which typically have very few nonlinear terms [12, 19]. The pooling problem arises in petroleum refining where various input streams are blended together to form a pool whose qualities depend on the amount of each input. The resulting optimization model minimizes a linear function subject to linear and bilinear constraints [8, 16]. Once again, the bilinear constraints contain relatively few nonlinear terms. In antenna array signal processing, the goal is to minimize undesired noise sources while preserving the desired signal. The corresponding optimization problem minimizes a convex quadratic function subject to several linear constraints and a few quadratic equality constraints of very low rank [23, 17]. Stackelberg games feature two players, a leader and a follower. The leader makes a decision in anticipation of the follower's reaction, while the follower's decision space depends on the leader's strategy. This can be modeled as a bilevel programming problem and, under certain conditions, is equivalent to minimizing a concave objective function subject to separable quadratic constraints [2].

Finding provably global solutions to problems of this sort is difficult. Provably convergent outer approximation algorithms can be applied to QCQP. However, thus far these methods are shown to be practical only for very small problem sizes [10]. The GOP algorithm, a decomposition approach which converts the original problem into primal and relaxed dual subproblems, has proven to be well-suited for quadratically constrained problems [26]. Another global optimization algorithm which is suitable for these problems is the Reformulation–Linearization Technique (RLT) [22]. RLT uses the upper and lower bounds on the decision variables to generate a linear relaxation of the original problem. A natural extension of this approach is to generate convex, rather than linear, relaxations [21, 20].

In this paper, we present a global optimization algorithm that relies on Lagrangian duality to generate lower bounds on the optimal objective function value. Lagrangian duality is a well-known optimization tool that can be employed in wide variety of contexts. Assuming that only the complex constraints of QCQP are dualized, its Lagrangian function is given by

$$\phi(x, v) = f^0(x) + \sum_{i=1}^{m} (v^i [f^i(x) - b_i]),$$

where v is the vector of dual variables. The Lagrangian dual problem of QCQP is

$$\max_{v \geqslant 0} \Theta(v) = \min_{x \in X} \{\phi(x, v)\},$$

where $X = \{x : \ell \leqslant x \leqslant u\}$. Any feasible dual solution $\hat{v}$ yields a lower bound of $\Theta(\hat{v})$ on z^*, the optimal objective function value of QCQP. Since QCQP is

nonconvex, $\Theta(v^*)$, the objective function value of the optimal dual solution v^*, will not necessarily equal z^*.

Two difficulties arise in attempting to use Lagrangian duality to solve a nonconvex problem. The first is the duality gap; solving the dual does not necessarily yield the primal solution. Given a nonconvex objective function ψ and linear constraints, Falk [7] showed that the solution of the Lagrangian dual problem yields the minimum of the convex envelope of ψ, rather than the minimum of ψ, over the constraint set. He suggests utilizing Lagrangian duality within a branch-and-bound framework that partitions the feasible region.

More recently, other researchers have extended this idea of combining Lagrangian duality with a partitioning strategy. Ben-Tal et al. [5] proved that making a sufficiently fine partition will ensure that the duality gap is less than any specified tolerance ϵ. Similarly, Dur and Horst [6] prove that, for very general classes of nonconvex programs, the duality gap obtained by solving the Lagrangian dual is reduced to zero in the limit when combined with a suitably refined partitioning of the feasible set. Barrientos and Correa [3] derive a similar result. In each paper, these results motivate a convergent branch-and-bound algorithm that uses Lagrangian duality to generate bounds.

While a suitable partitioning strategy can overcome the duality gap, the second difficulty with using Lagrangian duality for global optimization is that it requires the minimization of a nonconvex function. Thus, it typically is proposed for problems whose structure ensures a tractable minimization subproblem. The papers mentioned in the previous paragraph provide a nice illustration of this. Ben-Tal et al., solve pooling problems using an enumeration strategy that exploits the bilinear constraints of these problems. Dur and Horst [6] also consider bilinearly constrained problems, including nonconvex programs which can be transformed into this from. They also analyze concave minimization under reverse convex constraints, whose dual subproblems can also be solved by vertex enumeration. To minimize a general nonconvex function over a polytope, they use a convex envelope construction to ensure that the dual function generates a valid lower bound. Barrientos and Correa [3] transform quadratic programs so that their objective functions are separable. The Lagrangian subproblem thus reduces to the minimization of a separable quadratic function over variable bounds.

Additional applications of Lagrangian duality further underscore that its utility for global optimization often relies on exploiting problem structure to successfully minimize the nonconvex subproblem. Adhya et al. [1] prove that the constraint structure of the pooling problem implies that its Lagrangian subproblem will have an optimal solution in which every variable is at one of its bounds. This allows them to transform the nonconvex program into a mixed-integer program that is tractable for reasonably sized problems. Kuno and Utsunomiya [14] apply Lagrangian duality to production–transportation problems. They show that their dual problem can be broken down into m subproblems, each of which sums a concave function and n piecewise affine functions. These can be minimized by comparing function values

at the relevant break points in the affine functions. In the absence of a tractable nonconvex subproblem, Lagrangian relaxation can be used for global minimization either by applying it to a linearization or convexification of the original problem [21] or by developing a global search strategy that escapes local minimums [27].

Rather than finding the Lagrangian of the convexification of QCQP, we consider the convexification of the Lagrangian of QCPQ. Li and Sun [15] have also proposed utilizing a convexification of the Lagrangian. They consider the 'p-power' Lagrangian, which is the Lagrangian of the modified program

$$\min_{x \in X} \quad [f^0(x)]^p \qquad \text{s.t.} \quad [f^i(x)]^p \leqslant b_i^p, \ \forall i.$$

They show that, under certain conditions, the Hessian of the p-power Lagrangian is positive definite at a local minimum for sufficiently large p. They use this result to demonstrate the local optimality of a solution at which the Hessian of the Lagrangian is not positive definite.

The main advantage of the approach presented here is the relative ease with which the proposed convexification of the Lagrangian can be minimized over bound constraints. This allows it to be naturally incorporated into a branch-and-bound scheme. An exhaustive partitioning process guarantees that the convexification of the Lagrangian approaches the Lagrangian, so it is not surprising that algorithm can be shown to converge to the global solution.

In section two, convex relaxations of QCQP are compared to convex underestimates of the restricted Lagrangian. Section three describes a branch-and-bound algorithm which uses these convex underestimates to provide lower bounds on the problem's optimal objective function value within a given partition of the original feasible region. Section four provides computational results, comparing the performance of the algorithm presented in section three with a rudimentary implementation of RLT. Section five discusses relevant computational issues, and section six provides some initial conclusions and directions for additional research in this area.

2. Convex relaxations and convex underestimates

Several techniques can be used to generate convex relaxations of nonconvex programs [21, 20]. Here we restrict our discussion to convex relaxations which can be derived without introducing additional variables or constraints. Assuming Q^i is not positive semi-definite, let $\lambda^i_{min} < 0$ be its smallest eigenvalue. If γ is a constant that satisfies $\gamma + \lambda^i_{min} > 0$, then the matrix $\gamma I + Q^i$ is positive definite. Adding and subtracting $x^T(\gamma I)x$ to $f^i(x)$ allows $f^i(x)$ to be expressed as the difference of two convex functions:

$$f^i(x) = c_i^T x + x^T Q^i x = c_i^T x + x^T(\gamma I + Q^i)x - \gamma x^T x.$$

Variable bounds can be used to generate a linear underestimate of $-\gamma x^T x$:

$$\gamma(u-x)^T(x-\ell) \geqslant 0$$
$$\gamma(u^T x - u^T\ell - x^T x + x^T\ell) \geqslant 0$$
$$\gamma((u+\ell)^T x - u^T\ell) \geqslant \gamma x^T x - \gamma((u+\ell)^T x - u^T\ell) \leqslant -\gamma x^T x$$

Since $x^T(\gamma I + Q^i)x$ is convex, this inequality leads to a convex underestimate for $f^i(x)$:

$$f^i(x) = c_i^T x + x^T(\gamma I + Q^i)x - \gamma x^T x \geqslant c_i^T x + x^T(\gamma I + Q^i)x - \gamma((u+\ell)^T x - u^T\ell).$$

An alternative approach is to generate a convex underestimate for each individual nonconvex term [24]. Define the index sets $J_i^+ = \{j : q_{jj}^i \geqslant 0\}$, $J_i^- = \{j : q_{jj}^i < 0\}$, $K_i^+ = \{(j,k) : q_{jk}^i \geqslant 0\}$ and $K_i^- = \{(j,k) : q_{jk}^i < 0\}$. Using these sets, function $f^i(x)$ can be written in summation form:

$$f^i(x) = \sum_{j=1}^n c_j^i x_j + \sum_{J_i^+} q_{jj}^i x_j^2 + \sum_{J_i^-} q_{jj}^i x_j^2 + 2\sum_{K_i^+} q_{jk}^i x_j x_k + 2\sum_{K_i^-} q_{jk}^i x_j x_k$$
$$= \sum_{j=1}^n c_j^i x_j + \sum_{J_i^+} q_{jj}^i x_j^2 - \sum_{J_i^-} |q_{jj}^i| x_j^2 + 2\sum_{K_i^+} q_{jk}^i x_j x_k - 2\sum_{K_i^-} |q_{jk}^i| x_j x_k$$

Convex underestimates for each $x_j x_k$ term are generated by using two relationships:

$$2q_{jk}^i x_j x_k = q_{jk}^i(x_j + x_k)^2 - q_{jk}^i(x_j^2 + x_k^2) \tag{1}$$
$$-2|q_{jk}^i| x_j x_k = |q_{jk}^i|(x_j - x_k)^2 - |q_{jk}^i|(x_j^2 + x_k^2) \tag{2}$$

Substituting these equations into the expression for $f^i(x)$ yields

$$f^i(x) = \sum_{j=1}^n c_j^i x_j + \sum_{J_i^+} q_{jj}^i x_j^2 - \sum_{J_i^-} |q_{jj}^i| x_j^2 + \sum_{K_i^+} q_{jk}^i[(x_j + x_k)^2 - (x_j^2 + x_k^2)]$$
$$+ \sum_{K_i^-} |q_{jk}^i|[(x_j - x_k)^2 - (x_j^2 + x_k^2)].$$

To simplify this expression, let

$$p_j^i = \begin{cases} \sum_{k \neq j} |q_{jk}^i| + |q_{jj}^i|, & j \in J_i^- \\ \sum_{k \neq j} |q_{jk}^i|, & \text{otherwise.} \end{cases}$$

Now $f^i(x)$ can be expressed as

$$f^i(x) = \sum_{j=1}^{n} c_j^i x_j + \sum_{J_i^+} q_{jj}^i x_j^2 + \sum_{K_i^+} q_{jk}^i (x_j + x_k)^2 + \sum_{K_i^-} |q_{jk}^i| (x_j - x_k)^2 - \sum_{j=1}^{n} p_j^i x_j^2.$$

Using the inequality

$$(u_j - x_j)(x_j - \ell_j) \geqslant 0 \Longrightarrow (u_j + \ell_j)x_j - u_j \ell_j \geqslant x_j^2,$$

and the fact that $p_j^i \geqslant 0$, it follows that

$$-p_j^i x_j^2 \geqslant -p_j^i((u_j + \ell_j)x_j - u_j \ell_j).$$

It is easily shown that, for $q_{jk}^i \geqslant 0$, both $q_{jk}^i (x_j + x_k)^2$ and $|q_{jk}^i|(x_j - x_k)^2$ are convex. Thus, the following inequality provides a convex underestimate of $f^i(x)$:

$$f^i(x) \geqslant \sum_{j=1}^{n} ([c_j^i - p_j^i(u_j + \ell_j)]x_j + p_j^i \ell_j u_j) + \sum_{J_i^+} q_{jj}^i x_j^2 + \sum_{K_i^+} q_{jk}^i (x_j + x_k)^2 + \sum_{K_i^-} |q_{jk}^i| (x_j - x_k)^2 \tag{3}$$

This convex underestimate for the nonlinear term $x_j x_k$ has an intuitively appealing geometric interpretation. Consider the case where $0 \leqslant x_j, x_k \leqslant 10$. The term $x_j x_k = 0$ at the points $(x_j, x_k) = (0, 10)$ and $(x_j, x_k) = (10, 0)$. Thus, any convex underestimate of $x_j x_k$ must be $\leqslant 0$ for each point on the line between these two points, including the point $(x_j, x_k) = (5, 5)$.

With this in mind, consider the convex underestimate of $x_j x_k$ along the line $x_j = x_k$. At $(x_j, x_k) = (5, 5)$, the convex underestimate must be $\leqslant 0$. In order for the underestimate to be tight at the variable bounds, it must equal 0 at $(x_j, x_k) = (0, 0)$ and must equal 100 at $(x_j, x_k) = (10, 10)$. The convex underestimate proposed above,

$$x_j x_k \geqslant f(x_j, x_k) = \frac{1}{2}[(x_j + x_k)^2 - 10x_j - 10x_k],$$

is the quadratic curve that connects the three points $(f(x_j, x_k), x_j, x_k) = (0, 0, 0)$, $(0, 5, 5)$ and $(100, 10, 10)$ along the line $x_j = x_k$. This is illustrated in Figure 1. This figure also shows that a tighter underestimate is generated using the tangent lines to the curve $x_j x_k$ at the lower and upper bounds. This tighter underestimate is the one generated by RLT.

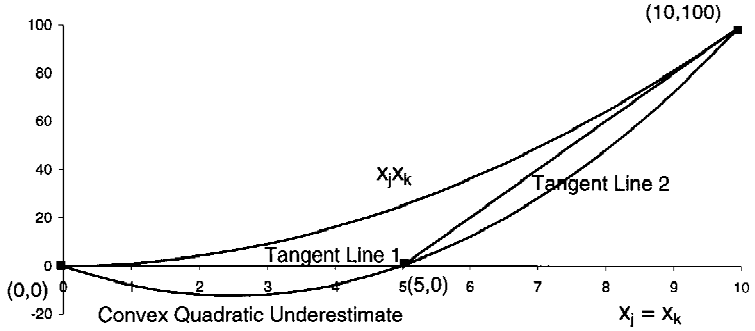

Figure 1. Convex underestimate of $x_j x_k$ along the line $x_j = x_k$.

The quality of the lower bound generated by these convex relaxations depends on the structure of the problem. It generates the convex envelope of concave terms and easily incorporates convex terms, which a linearization approach such as RLT relaxes. (Certainly RLT can be extended to retain convex terms [21], but linear relaxations remain an attractive alternative because of the availability and quality of linear programming software.) Since RLT provides the convex envelope of a bilinear term over a box, it will always dominate any other convex relaxation of these terms. In this case, the only advantage of the convex relaxations given above is that they do not require any additional variables or constraints.

On the other hand, if a problem contains both convex and nonconvex terms, convex underestimates of the Lagrangian can yield a significantly better bound than RLT, even if RLT is extended to retain convex terms. As an illustration, consider the following problem:

$$\begin{array}{lll} \text{(EX1)} & \text{Min} & 6x_1^2 + 4x_2^2 + 5x_1x_2 \\ & \text{subject to} & -6x_1x_2 \leqslant -48 \\ & & 0 \leqslant x_1,\ x_2 \leqslant 10 \end{array}$$

The optimal solution to EX1 is $(x_1, x_2) = (2.5558, 3.1302)$, which yields an objective function value of 118.384. Using equation (3) to generate convex underestimates of the objective and constraint functions over the region defined by the upper and lower bounds on x_1 and x_2 yields a convex program:

$$\begin{array}{ll} \text{Min} & 6x_1^2 + 4x_2^2 + 2.5(x_1 + x_2)^2 - 2.5(10x_1 + 10x_2) \\ \text{s.t.} & 3(x_1 - x_2)^2 - 3(10x_1 + 10x_2) \leqslant -48 \\ & 0 \leqslant x_1, x_2 \leqslant 10 \end{array}$$

The optimal solution of this convex program, $(x_1, x_2) = (1.02, 1.53)$, yields a relatively poor lower bound of -31.9 on the optimal objective function value of EX1.

A much better lower bound of 46.4 is generated by extending RLT to include convex terms:

$$\begin{aligned} \text{Min} \quad & 6x_1^2 + 4x_2^2 + 5w_{12} \\ \text{s.t.} \quad & -6w_{12} \leqslant -48 \\ & \max\{0, 10x_1 + 10x_2 - 100\} \leqslant w_{12} \leqslant \min\{10x_1, 10x_2\} \\ & 0 \leqslant x_1, x_2 \leqslant 10. \end{aligned}$$

The solution to this convex program is $(x_1, x_2, w_{12}) = (\frac{4}{5}, \frac{4}{5}, 8)$.

For this problem, convex underestimates of the Lagrangian can yield a much better bound than RLT. By dualizing only the nonlinear constraint, the Lagrangian dual problem of EX1 is

$$\max_v \ \Theta(v), \quad \text{where}$$
$$\Theta(v) = \inf_x \{6x_1^2 + 4x_2^2 + 5x_1x_2 + v(48 - 6x_1x_2) : 0 \leqslant x_1, x_2 \leqslant 10\}.$$

Any value of $v \geqslant 0$ provides a lower bound on z^*; for example, if $v = \frac{7}{3}$, the restricted Lagrangian $L(x) = \phi(x, \frac{7}{3})$ is minimized at $(x_1, x_2) = (0, 0)$ with $\Theta(\frac{7}{3}) = 112$. Equation (3) provides a convex underestimate $\hat{L}(x)$ of $L(x) = \phi(x, \frac{7}{3})$:

$$\begin{aligned} L(x) &= 6x_1^2 + 4x_2^2(5 - \frac{7}{3} \times 6)x_1x_2 + \frac{7}{3} \times 48 \\ &= 6x_1^2 + 4x_2^2 + \frac{9}{2}(x_1 - x_2)^2 - \frac{9}{2}(x_1^2 + x_2^2) + 112 \\ &= \frac{3}{2}x_1^2 - \frac{1}{2}x_2^2 + \frac{9}{2}(x_1 - x_2)^2 + 112 \\ L(x) \geqslant \hat{L}(x) &= \frac{3}{2}x_1^2 - \frac{1}{2}(10x_2)\frac{9}{2}(x_1 - x_2)^2 + 112 \end{aligned}$$

Within the region $0 \leqslant x_1, x_2 \leqslant 10$, $\hat{L}(x)$ is minimized at $(x_1, x_2) = (\frac{5}{3}, \frac{20}{9})$, at which point $\hat{L}(x) = 106\frac{4}{9}$. Table 1 gives the values of x_1 and x_2 that minimize $\hat{L}(x)$ over X and the corresponding value of $\hat{L}(x)$ as functions of v.

The formulas in Table 1 imply that the convex underestimate of the Lagrangian generates a better bound than RLT for all $v \in [0.967, 2.621]$.

Convex underestimates of the Lagrangian can be seen to offer significant advantages over other approaches. Like the convex relaxations, it does not require additional variables or constraints, as does RLT. In many instances, RLT will yield better bounds than convex underestimates of the Lagrangian, but, as EX1 demonstrates, the underestimates can also yield better bounds than RLT. Obviously, minimizing the Lagrangian function itself would provide better bounds than an underestimating function. However, efficient nonlinear programming methods for

Table 1. Bounds generated for various values of v

v	x_1	x_2	$\hat{L}(x)$
$0 \leqslant v \leqslant \frac{13}{6}$	0	0	$48v$
$\frac{13}{6} \leqslant v \leqslant 2.5914$	$\frac{60(6v-13)}{(6v-5)(17-6v)}$	$\frac{5(6v-13)}{(17-6v)}$	$48v - \frac{150(6v-13)^2}{(17-6v)(6v-5)}$
$2.5914 \leqslant v \leqslant \frac{17}{6}$	$\frac{5}{6}(6v-5)$	10	$295\frac{5}{6} + 298v - 150v^2$
$v \geqslant \frac{17}{6}$	10	10	$1500 - 552v$

minimizing nonconvex functions do not necessarily identify the global minimum. Using convex underestimates eliminates this concern and thus allows Lagrangian duality to be naturally incorporated into a global optimization framework.

3. A Branch-and-Bound Algorithm

A branch-and-bound algorithm was developed that relied on convex underestimates of the Lagrangian to generate lower bounds on the optimal objective function value. This algorithm integrates three techniques for solving nonlinear problems.

(1) Lagrangian underestimates are used to provide lower bounds.
(2) Newton's method is used to obtain local solutions to the primal problem.
(3) The interval Newton method is used to facilitate convergence to the global solution.

As its name indicates, the interval Newton method is similar to Newton's method for solving systems of equations. While Newton's method begins with a vector of values x_k and finds a new vector of values x_{k+1}, the interval Newton method begins with a vector of intervals X_k and finds a new vector of intervals X_{k+1}. A full description of the method can be found in [11]. Like Newton's method, the interval Newton method does not always converge. However, unlike Newton's method, when the interval Newton method is successful, it allows a strong conclusion to be made regarding the original bounded region X_0.

(1) If $X_{k+1} \cap X_k = \emptyset$, then X_0 does not contain any solution to the given system of equations.
(2) If the sequence of interval vectors collapses to a single point, then that point is the only solution to the given system of equations within X_0.

The algorithm presented below will subsequently be referred to as the LagInt algorithm. Notationally, B^i is the hyperrectangular region defined by upper and lower bound vectors u^i and ℓ^i. Let lb^i be the lower bound on the optimal objective

function value within B^i, and v^i be the current estimate of the Lagrange multipliers within B^i. Let lb and ub be the overall lower and upper bounds on the problem's optimal objective function value. Since ℓ and u are finite, an initial value for lb can be computed using interval arithmetic. No true upper bound can be computed, since any given problem instance could be infeasible. However, we can compute an upper bound on the value of the objective function $f^0(x)$ over B^1. This number is useful, since it allows us to fathom infeasible regions.

LAGINT ALGORITHM FOR SOLVING QCQP

Initialization Let $\ell^1 = \ell$ and $u^1 = u$, $S = \{B^1\}$, $v^1 = 0$, Compute initial values for lb and ub using interval arithmetic. Set cnt to some positive integer.

Iteration k

(1) $j \leftarrow \arg\min_i \{lb^i : B^i \in S\}$.

(2) Find lower bounds for region B^j:

$$h \leftarrow 1, \quad v^1 \leftarrow v^j, \text{ and } z^0 = lb^j.$$

while $h \leqslant cnt$, $lb^j < ub$ and $z^h > z^{h-1}$ do

$$L(x) = f^0(x) + \sum_{i=1}^{m} v_i^h (f^i(x) - b_i)$$

and $\hat{L}(x)$ be a convex underestimate of $L(x)$.

$x^h \leftarrow \arg\min\{\hat{L}(x) : x \in B^i\}$ and $z^h \leftarrow \min\{\hat{L}(x) : x \in B^i\}$

If $h = 1$ or $z^h > lb^j$, $\hat{x} \leftarrow x^h$ and $\hat{v} \leftarrow v^h$.

If $z^h > lb^j$, $lb^j \leftarrow z^h$

If $lb^j > ub$, then $S \leftarrow S \setminus B^j$ and go to 6.

If x^h is feasible and $f^0(x^h) \leqslant ub$, $ub \leftarrow f^0(x^h)$ and update incumbent solution

$$\xi_i^h = \begin{cases} f^i(x^h), & f^i(x^h) > 0 \text{ or } v_i^h > 0 \\ 0, & \text{otherwise.} \end{cases}$$

If $h = 1$, $d^h = \xi^h$, else $d^h = \xi^h + \frac{\|\xi^h\|}{\|d^{h-1}\|} d^{h-1}$.

$v_i^{h+1} = \max\{0, v_i^h + \lambda_h d_i^h(x)\}$, where λ_h is the step-size parameter.

$h \leftarrow h + 1$

(3) Perform range reduction tests [18] to decrease the width of the interval $[\ell_j, u_j]$ for each x_j.

(4) Using $\hat{x}$ and $\hat{v}$ as primal and dual starting points, apply Newton's method to solve the Karush–Kuhn Tucker necessary conditions. If a feasible solution

is found with a better objective function value than the incumbent solution, update ub and the incumbent solution.

(5) Apply the interval Newton method to find all solutions to the Fritz-John optimality conditions contained in B^j. If the interval Newton method identifies a feasible solution with a better objective function value than the incumbent solution, update ub and the incumbent solution. If the interval Newton method converges to a single point or proves that no solution to the optimality conditions exists within B^j, then $S \leftarrow S \setminus B^j$.

Otherwise partition B^i into two subregions: B^{2k} and B^{2k+1} using any exhaustive partitioning process. Let $lb^{2k} = lb^{2k+1} = lb^j$ and $v^{2k} = v^{2k+1} = v^h$.

$$S \leftarrow S \setminus B^j \cup \{B^{2k}, B^{2k+1}\}$$

(6) If $S = \emptyset$, stop. If an incumbent solution has been found, then the algorithm has found an optimal solution. Otherwise, the problem is infeasible. If $|S| \geqslant 1$, goto 1.

PROPOSITION 3.1. *Given a feasible instance of QCQP, if the algorithm terminates, it terminates with the global optimal solution.*

Proof. If $S = \emptyset$, then one of the following three statements must be true for each region B^j at the bottom of the branch-and-bound tree.

(1) The Interval Newton method demonstrates that there are no points within the region that satisfy necessary conditions for optimality.

(2) The Interval Newton method identifies the only point within the region which satisfies the necessary conditions.

(3) The lower bound generated by the convex underestimate of the Lagrangian, lb^j, is greater than ub. If an incumbent solution has been found, then lb^j is greater than the objective function value of a feasible solution to the primal problem. Otherwise, the lower bound lb^j is greater than the largest possible objective function value of any feasible solution.

If an incumbent solution exists, its optimality follows immediately. Otherwise, the problem must be infeasible. □

If the algorithm does not terminate in a finite number of iterations, then a slightly modified version of the given algorithm can be shown to converge to the optimal solution. The globally convergent version of the algorithm differs from the given LagInt algorithm in two particulars:

(1) $cnt = 1$, which implies that $d^h = \xi^h, \quad \forall\ h$.

(2) For each region B^j, let t^j be its level within the branch-and-bound tree (i.e. if B^j has three ancestors, then $t^j = 4$). Make $\lambda_h \leftarrow \frac{1}{t^j}$ and

$$v^{h+1} = P_{v \geqslant 0}[v^h + \lambda_h \frac{d^h}{\| d^h \|}],$$

where $P_{v \geqslant 0}$ is the projection of $v^h + \lambda_h \frac{d^h}{\|d^h\|}$ onto $\{v : v \geqslant 0\}$.

This modification maximizes the dual function using a step-size chosen to satisfy the well-known divergent series rule (i.e. $\lambda^h \to 0$ and $\sum_{h=1}^{\infty} \lambda_h = \infty$). First, we show that the convex underestimates of the Lagrangian converge to the Lagrangian.

LEMMA 3.1. *Let ψ be any infinite path down the branch-and-bound tree, and let $\{v^{\psi(k)}\}_{k=1}^{\infty}$ be the infinite sequence of dual solutions found for nested partitions $B^{\psi(1)} \supset B^{\psi(2)} \supset B^{\psi(3)} \ldots$ along path ψ. If $L^{\psi(k)}(x)$ is the Lagrangian function given multipliers $v^{\psi(k)}$ and $\hat{L}^{\psi(k)}(x)$ is the convex underestimate developed using the method described in section two, then* $\lim_{k\to\infty}(L^{\psi(k)}(x) - \hat{L}^{\psi(k)}(x)) = 0$.

After removing common terms, we have

$$\begin{aligned} L^{\psi(k)}(x) - \hat{L}^{\psi(k)}(x) &= -\sum_{j=1}^{n} p_j x_j^2 - (-\sum_{j=1}^{n} p_j[(\ell_j^{\psi(k)} + u_j^{\psi(k)})x_j - \ell_j^{\psi(k)} u_j^{\psi(k)}]) \\ &= \sum_{j=1}^{n} p_j([\ell_j^{\psi(k)} + u_j^{\psi(k)}]x_j - \ell_j^{\psi(k)} u_j^{\psi(k)} - x_j^2) \\ &= \sum_{j=1}^{n} p_j(x_j - \ell_j^{\psi(k)})(u_j^{\psi(k)} - x_j) \end{aligned}$$

The exhaustive partitioning process ensures that $\lim_{k\to\infty}(u_j^{\psi(k)} - \ell_j^{\psi(k)}) = 0$. This implies that $\lim_{k\to\infty}(x_j - \ell_j^{\psi(k)}) = \lim_{k\to\infty}(u_j^{\psi(k)} - x_j) = 0,\ \forall\, x_j \in [\ell^{\psi(k)}, u^{\psi(k)}]$ and thus that $\lim_{k\to\infty}(L^{\psi(k)}(x) - \hat{L}^{\psi(k)}(x)) = 0$. □

Next, we characterize the limit point of the sequence of dual solutions.

LEMMA 3.2. *Let $\lim_{k\to\infty} v^{\psi(k)} = \bar{v}^{\psi}$. Let $\{x^{\psi(k)}\}_{k=1}^{\infty}$ be the sequence of primal solutions along path ψ, with $\bar{x}^{\psi} = \lim_{k\to\infty} x^{\psi(k)}$. If $f^i(\bar{x}^{\psi}) < b_i$, then $\bar{v}_i^{\psi} = 0$.*

Let $\varepsilon_i = b_i - f^i(\bar{x}^{\psi}) > 0$ and let α be any constant satisfying $0 < \alpha < 1$. Since $f^i(x)$ is continuous and $\lim_{k\to\infty}(u^{\psi(k)} - \ell^{\psi(k)}) = 0$, there will eventually be some $\hat{k}$ such that $b_i - f^i(x) \geqslant \alpha\varepsilon_i,\ \forall x \in B^{\psi(k)},\ k \geqslant \hat{k}$. Thus, $v_i^{\psi(\hat{k}+1)} \leqslant \max\{0, v^{\psi(\hat{k})} - \lambda_{\psi(\hat{k})}(\alpha\varepsilon_i)\}$ and $v_i^{\psi(\hat{k}+r)} \leqslant \max\{0, v^{\psi(\hat{k})} - (\alpha\varepsilon_i)(\sum_{j=0}^{r-1} \lambda_{\psi(\hat{k}+j)})\}$. Since $\sum_{j=0}^{\infty} \lambda_{\psi(\hat{k}+j)} = \infty$, it is clear that $\lim_{k\to\infty} v_i^{\psi(k)} = 0$. □

Third, we show that the limit point of any infinite subsequence is feasible.

LEMMA 3.3. *The limit point $\bar{x}^{\psi}$ of any infinite subsequence $\{x^{\psi(k)}\}_{k=1}^{\infty}$ is feasible.*

Proof. Assume for the sake of contradiction that the limit point $\bar{x}^{\psi}$ of an infinite sequence is infeasible. This means that there is some constraint i for which $f^i(\bar{x}^{\psi}) - b_i = \varepsilon_i > 0$. Once again, given some constant $\alpha \in (0, 1)$, the continuity of $f^i(x)$ and the exhaustive partitioning process ensure that there is some $\hat{k}$ for which $f^i(x) - b_i \geqslant \alpha\varepsilon_i,\ \forall x \in B^{\psi(k)},\ k \geqslant \hat{k}$. Thus, the dual solution

$v_i^{\psi(\hat{k}+r)} \geqslant v^{\psi(\hat{k})} + (\alpha\varepsilon_i)(\sum_{j=0}^{r-1} \lambda_{\psi(\hat{k}+j)})$. Since $\sum_{j=0}^{\infty} \lambda_{\psi(\hat{k}+j)} = \infty$, it is clear that $\lim_{k\to\infty} v_i^{\psi(k)} = \infty$. Lemma 3.2 ensures that as $k \to \infty$, no constraint which is satisfied with slack has a nonzero multiplier. This implies that $\lim_{k\to\infty} L^{\psi(k)} \geqslant \lim_{k\to\infty}(f^0(x^{\psi(k)}) + v_i^{\psi(k)}[f^i(x^{\psi(k)}) - b_i])$. Since $f^0(x)$ is bounded, $f^i(\bar{x}^\psi) - b_i = \varepsilon_i$ and $\lim_{k\to\infty} v_i^{\psi(k)} = \infty$, then $\lim_{k\to\infty} L^{\psi(k)} = \infty$. Lemma 3.1 implies that $\lim_{k\to\infty} \hat{L}^{\psi(k)} = \lim_{k\to\infty} L^{\psi(k)} = \infty$. However, this is a contradiction, since the region $B^{\psi(k)}$ would be fathomed when its lower bound exceeded the value of ub. Thus, there cannot be an infinite subsequence whose limit point is infeasible. □

PROPOSITION 3.2. *The limit point $\bar{x}^\psi$ of any infinite subsequence is a global optimal solution.*

Proof. Lemma 3.3 ensures that $f^i(\bar{x}^\psi) \leqslant b_i,\ i = 1, \ldots, m$. Hence, $f^0(\bar{x}^\psi) \geqslant z^*$. Since the region with the minimum lower bound is selected to be explored during each iteration, $\hat{L}^{\psi(k)} \leqslant z^*,\ \forall\ k$. Combining this with Lemma 3.1 yields $\lim_{k\to\infty} L^{\psi(k)} = \lim_{k\to\infty} \hat{L}^{\psi(k)} \leqslant z^*$. As $k \to \infty$, Lemma 3.2 ensures that complementary slackness will hold and hence $\lim_{k\to\infty} L^{\psi(k)} = f^0(\bar{x}^\psi)$. Thus, $f^0(\bar{x}^\psi) = z^*$ and $\bar{x}^\psi$ is a global optimal solution. □

4. Algorithmic considerations

While choosing λ using the divergent series rule is theoretically attractive, it tends to converge slowly. Thus, in the spirit of conjugate gradient methods, the LagInt algorithm uses an average direction strategy for which d^h bisects d^{h-1} and ξ^h [4]. A theoretically attractive formula for step-size λ_h is $\lambda_h = \frac{\beta_h}{\|\xi^h\|}(\Theta(v^*) - \Theta(v^h))$, where β_h is some small constant [4].

Since the value of $\Theta(v^*)$ is not known, different approaches were used to develop an estimate $\bar{\Theta}(v^*)$ of $\Theta(v^*)$. The one used in the test problems was

$$\bar{\Theta}(v^*) = \hat{L}(x^k) + \sum_{j=1}^{n}([u_j + \ell_j]x_j^k - u_j\ell_j - (x_j^k)^2),$$

where $x^k = \arg\min\{\hat{L}(x) : x \in B^k\}$.

After an upper bound ub on the optimal primal objective function value had been found, this estimate $\bar{\Theta}(v^*)$ was updated using the formula $\bar{\Theta}(v^*) \leftarrow \max\{\bar{\Theta}(v^*), \frac{1}{2}\bar{\Theta}(v^*) + \frac{1}{2}ub\}$.

In Section 2, two methods were discussed for generating a convex underestimate of a quadratic function. For the test problems, convex underestimates were created by adding the quadratic form $x^T Px$ to restricted Lagrangian $L(x) = b + c^t x + x^T Qx$, where P is a diagonal matrix with constants $p_j = \max\{0, \sum_{k\neq j} q_{jk} + \epsilon\}$ along the diagonal. Here ϵ is a positive constant that makes $Q + P$ positive definite. In computational tests, the value $\epsilon = 0.25$ was used. A linear underestimate of

the concave term $-x^T Px$ is then subtracted from $L(x)$ to complete the convex underestimate.

For example, let the restricted Lagrangian function be given by

$$L(x) = 30 + [1, 4, 2]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [x_1, x_2, x_3]\begin{bmatrix} 3 & 2 & 1 \\ 2 & 7 & 4 \\ 1 & 4 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

with $2 \leqslant x_1, x_2, x_3 \leqslant 10$.

Using the formula given above,

$$P = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2.25 \end{bmatrix}$$

and the convex underestimate of $L(x)$ is

$$\begin{aligned} \hat{L}(x) =& 30 + [1, 4, 2]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [x_1, x_2, x_3]\begin{bmatrix} 3.25 & 2 & 1 \\ 2 & 7 & 4 \\ 1 & 4 & 5.25 \end{bmatrix} \\ & - 0.25(12x_1 - 20) - 2.25(12x_3 - 20) \\ =& 80 + [-2, 4, -25]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [x_1, x_2, x_3]\begin{bmatrix} 3.25 & 2 & 1 \\ 2 & 7 & 4 \\ 1 & 4 & 5.25 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{aligned}$$

Another issue is how to best exploit the simple structure of the variable bound constraints. Initial values for all the dual variables (including those associated with variable bounds) are used to create the Lagrangian underestimate $\hat{L}(x)$. Given these initial values, the algorithm attempts to solve the convex program $\min_x\{\hat{L}(x) : \ell_j \leqslant x_j \leqslant u_j,\ j = 1, \ldots, n\}$ by making a series of conjectures regarding which constraints are tight. To illustrate this, consider the previous Lagrangian underestimate:

$$\hat{L}(x) = 80 + [-2, 4, -25]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [x_1, x_2, x_3]\begin{bmatrix} 3.25 & 2 & 1 \\ 2 & 7 & 4 \\ 1 & 4 & 5.25 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Initially, we conjecture that the variable bound constraints are tight if and only if their corresponding multipliers are nonzero. To illustrate the procedure, assume that all the multipliers associated with bound constraints are equal to zero. In this case, we conjecture that none of the bound constraints will be binding and minimize $\hat{L}(x)$. The point which minimizes this function is $x^1 = (0.8343, -3.1770, 4.6426)$; this point yields the value $\hat{L}(x^1) = 14.7790$.

This point is infeasible since both x_1^1 and x_2^1 are < 2. Thus, the new conjecture is that the constraints $x_1 \geqslant 2$ and $x_2 \geqslant 2$ are tight at the optimal solution, thereby

allowing the dual variables associated with these constraints, $v_{\ell 1}$ and $v_{\ell 2}$, to be nonzero. Given the conjecture that $x_1 = x_2 = 2$, new primal and dual solutions are found using the necessary conditions for optimality:

$$\nabla \hat{L}(x) = \begin{bmatrix} -2 + 6.5x_1 + 4x_2 + 2x_3 \\ 4 + 4x_1 + 14x_2 + 8x_3 \\ -25 + 2x_1 + 8x_2 + 10.5x_3 \end{bmatrix} = \begin{bmatrix} 19 + 2x_3 \\ 40 + 8x_3 \\ -5 + 10.5x_3 \end{bmatrix} = \begin{bmatrix} v_{\ell 1} \\ v_{\ell 2} \\ 0 \end{bmatrix}.$$

The solution x^2, v^2 to this system is $x_3 = 0.4762$, $v_{\ell 1} = 19.95$ and $v_{\ell 2} = 43.81$. The feasible dual solution generates a much better lower bound on $\min\{\hat{L}(x) : 2 \leqslant x_j \leqslant 10, \quad j = 1, \ldots, n\}$: $\hat{L}(x^2) = 139.81$.

Since $x_3^2 < 2$, a new conjecture is that $v_{\ell 3}$ should also be nonzero and $x_1 = x_2 = x_3 = 2$. Optimality conditions are

$$\nabla \hat{L}^2(x) = \begin{bmatrix} -2 + 6.5x_1 + 4x_2 + 2x_3 \\ 4 + 4x_1 + 14x_2 + 8x_3 \\ -25 + 2x_1 + 8x_2 + 10.5x_3 \end{bmatrix} = \begin{bmatrix} 23 \\ 56 \\ 16 \end{bmatrix} = \begin{bmatrix} v_{\ell 1} \\ v_{\ell 2} \\ v_{\ell 3} \end{bmatrix}.$$

Both the primal solution $x^3 = (2, 2, 2)$ and the dual solution $(v_{\ell 1}, v_{\ell 2}, v_{\ell 3})$ are feasible. Thus, x^3 minimizes $\hat{L}(x)$ over $\{x : 2 \leqslant x \leqslant 10\}$, with $\hat{L}(x^3) = 152$.

This approach requires an accurate conjecture regarding which bound constraints are binding. Although there were occasions when no such conjecture was forthcoming, these occasions were rare, especially after the early iterations of the algorithm. In most cases, this technique provided a computationally efficient technique for solving $\min\{\hat{L}(x) : \ell \leqslant x \leqslant u\}$.

5. Computational Results

To test the effectiveness of the LagInt algorithm, a series of test problems were solved of the form

$$\begin{aligned} \text{Min} \quad & f^0(x) = \sum_{j=1}^{n} (c_j^0 x_j + q_j^0 x_j^2) + \sum_{j=1}^{\frac{n}{2}} r_j^0 x_j x_{j+\frac{n}{2}} \\ \text{subject to} \quad & f^i(x) = \sum_{j=1}^{n} (c_j^i x_j + q_j^i x_j^2) + \sum_{j=1}^{\frac{n}{2}} r_j^i x_j x_{j+\frac{n}{2}} \leqslant b_i, \quad i = 1, \ldots, m \\ & 0 \quad \leqslant \quad x_j \leqslant 10, \quad j = 1, \ldots, n. \end{aligned}$$

All objective function coefficients c_j^0, q_j^0 and r_j^0 were randomly generated integers between 1 and 10; all constraint coefficients were randomly generated integers between -1 and -10. Right-hand side values b_i were randomly generated integers between -60 and -180.

The intent was to generate test problems that were nonconvex, yet, because they contained relatively few nonlinear terms of the form $x_j x_k$, $j \neq k$, possessed a

favorable structure for the LagInt algorithm. While this structure permits a convex objective function (although this was not usually the case), each of the nonlinear constraints is nonconvex. By construction, these problems are feasible. Furthermore, their optimal solutions tended to have relatively few nonzero variables.

For the test problems, the LagInt algorithm was compared to a rudimentary implementation of the Reformulation–Linearization Technique (RLT). The RLT procedure implements a branch-and-bound algorithm that uses the RLT linear relaxation to generate bounds at each node in the tree. The software package CPLEX 5.0 (the Windows NT version) was used to solve the linear subproblems required by RLT. Solutions to the RLT linear programs were considered feasible to the original problem if each replacement variable was within 0.0001 of their associated nonlinear term and no constraint was violated by more than 0.0001. This worth noting since the time required by RLT depends on this tolerance. Since the LagInt algorithm relies on the interval Newton method to identify optimal solutions and fathom the regions containing them, tolerances were not an important issue for its speed of convergence. All computational tests were performed using a Hewlett Packard Kayak XU computer with 64 MB of RAM.

Table 2 shows the average time and the average number of iterations required by both algorithms to solve 10 test problems with a given number of variables (N) and constraints (M). For both RLT and LagInt, each iteration corresponds to exploring a single node in the branch-and-bound tree by generating a bound over a given region B^j.

The LagInt algorithm performs extremely well on the smaller test problems. For the larger test problems, the two algorithms are quite competitive. Overall, the results demonstrate that LagInt is able to solve certain favorably structured problems with upto 30 variables and eight constraints in a reasonable amount of time. There were no test problems where the algorithm failed due to storage limitations or any other cause.

Table 3 indicates how CPU time required by LagInt was allocated between its various key subroutines. Typically about half the time is spent in solving the Lagrangian subproblems for computing lower bounds. A substantial portion of LagInt's computational time was spent implementing the Interval Newton method. This reflects the method's struggle to converge for many of the test problems. When this occurred, the majority of subroutine calls failed to generate a useful result.

LagInt's generally strong performance is somewhat surprising in light of the poor initial lower bounds that it generated at the root node, shown in Table 4. In 137 out of the 160 test problems, this lower bound was negative, and it was better than that obtained using RLT in only four test problems.

Despite the poor initial bounds generated by Lagrangian underestimates, the number of nodes required by the LagInt algorithm was typically similar to that required by RLT. This is due in part to the interval Newton method. However, it also indicates that the lower bounds generated by convex underestimates improve quickly. Test problem 1D, a problem with $N = 6$, $M = 4$ and $Z^* = 55.24$,

Table 2. Average times required by RLT and LagInt algorithms

		Average CPU (s)		Average no. of Iterations	
N	*M*	RLT	LagInt	RLT	LagInt
6	4	4.23	0.16	137.4	108.2
10	4	5.29	0.61	167.0	151.6
14	4	26.07	2.09	791.4	230.0
16	4	2.88	1.85	85.6	133.6
18	4	31.11	7.10	882.0	271.6
20	4	27.35	22.20	796.0	641.0
24	4	23.52	21.21	650.4	429.8
30	4	88.40	125.36	2231.6	1418.6
6	8	4.09	0.19	134.6	95.6
10	8	4.66	0.79	147.8	151.0
14	8	34.40	5.58	1012.2	488.4
16	8	12.22	3.68	364.2	308.0
18	8	9.48	16.34	276.6	615.2
20	8	33.94	17.88	952.0	550.2
24	8	16.26	17.39	443.2	404.4
30	8	645.2	296.2	12345	2314

illustrates this improvement. The root node lower bound obtained by RLT was 24.4, compared to a bound of -93.7 obtained by LagInt using Lagrangian underestimates. However, as we move down the branch-and-bound tree along the path whose subregions contain the optimal solution, the LagInt bounds compare favorably with the RLT bounds for nodes at the same level in the tree. This is pictured in Figure 2. Branching decisions are based on the relaxation's solution, so the two subregions are different for each pair of nodes at the same level in their respective trees. Hence, this comparison does not show that LagInt is generating superior bounds for a given region. Rather, it illustrates that LagInt's poor initial bounds do not necessarily imply poor lower bounds throughout the branch-and-bound tree.

While no general conclusions can be drawn from a single test problem, the entire set of test problems shows that LagInt can be effective despite poor initial bounds. This could indicate that the behavior pictured in Figure 2 is not atypical. If so, this could be because the solutions to the LagInt relaxations generate more efficient branching decisions than the solutions to the RLT relaxations. This possibility is supported by considering the solutions to the initial relaxations of the ten smallest test problems ($N = 4$ and $M = 6$). For these problems, the average Euclidean distance between a given problem's optimal solution and the solution to its initial RLT linear relaxation is 5.8, compared to an average distance of 2.5

Table 3. Breakdown of computational time expended by LagInt algorithm

		Total CPU Seconds	Percentage of CPU Time Consumed by Implementing		
N	*M*	(all test problems)	Lagrangian Bounds	Interval Newton	Newton's Method
6	4	1.6	65	25	10
10	4	6.1	55	39	5.4
14	4	21	54	38	7.8
16	4	19	54	30	16
18	4	71	42	50	7.8
20	4	222	52	42	6.5
24	4	212	45	50	5.6
30	4	1254	52	44	3.9
6	8	1.9	66	21	13
10	8	7.9	57	29	14
14	8	56	45	47	7.4
16	8	37	61	22	17
18	8	163	53	40	7.6
20	8	179	42	50	7.9
24	8	174	48	37	15
30	8	2962	41	56	2.7

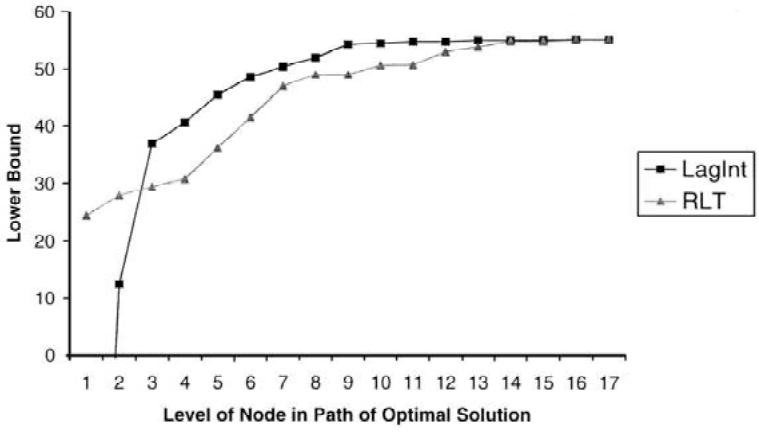

Figure 2. Lower bounds generated by LagInt and RLT.

Table 4. Quality of initial lower bound at root node

		Average root node $= (Z^* - LB)/Z^*$	
N	M	RLT	LagInt
6	4	28 %	157 %
10	4	30 %	140 %
14	4	35 %	211 %
16	4	28 %	219 %
18	4	29 %	275 %
20	4	30 %	313 %
24	4	30 %	296 %
30	4	37 %	326 %
6	8	31 %	99 %
10	8	28 %	180 %
14	8	37 %	189 %
16	8	25 %	248 %
18	8	32 %	202 %
20	8	33 %	235 %
24	8	26 %	268 %
30	8	33 %	379 %

between optimal solutions and the points which minimize the underestimate of the Lagrangian relaxation. The relative proximity of the optimal solutions and the solutions to the initial LagInt relaxation does not necessarily imply better branching decisions, but it would intuitively appear to be advantageous. This behavior could be a result of the structure and parameters of the test problems.

The test problems sought to answer whether or not Lagrangian underestimates could generate sufficiently tight bounds to enable a branch-and-bound algorithm to solve nonconvex programs in a reasonable amount of time. When used in conjunction with other techniques for facilitating convergence, the results indicate that they can, at least for problems of the given structure.

6. Conclusions and future work

The results presented in the previous section are by no means intended to demonstrate that LagInt is the best algorithm available for solving QCQP. In other tests, LagInt did not perform as well as RLT on problems with less favorable structure. Intuitively, LagInt would seem to be an attractive approach for nonconvex

programs whose quadratic forms are defined by relatively sparse matrices. This intuition is supported by its strong performance on the test problems, since they would be included in this category. Additionally, LagInt is attractive because it does not require the solution of linear programming subproblems.

Several directions are available for additional research:

(1) Improving the LagInt algorithm.

(2) Analyzing and applying the LagInt algorithm to new problems.

(3) Integrating LagInt with more established approaches such as RLT.

Initial work has demonstrated the viability of a global optimization algorithm that employs Lagrangian underestimates to generate lower bounds on the optimal objective function value. Additional work could provide greater insight with respect to computational issues that impact the efficiency of the algorithm.

Of all the implementation issues that could be addressed, one that stands out is the method used to update the dual variables. Using deflection directions proved to be an improvement over subgradients. Sophisticated techniques that have been applied to Lagrangian dual problems could prove to provide further advantages. Dual ascent algorithms have been successful for certain kinds of problems [25]. This approach is considered to be especially suitable for branch-and-bound, but must be tailored to the individual application [9]. Bundle methods have also worked well, and in some cases have been used along with subgradient directions [13].

Secondly, additional understanding of the strengths and weaknesses of the LagInt algorithm would help to determine in which contexts this approach could best be employed. Nonconvex problems with special structures, such as those mentioned in the first section, would seem to be the most suitable types of applications for the LagInt algorithm.

Finally, as more is learned about this approach, it could be used in connection with other methods within a branch-and-bound framework. Typically, it requires several iterations to establish good estimates of the Lagrange multipliers. Thus, in the early stages of the branch-and-bound tree, the algorithm generates relatively poor lower bounds. Perhaps Lagrangian underestimates could be strategically employed at later stages of the branch-and-bound process, when good multiplier estimates have been obtained via RLT or some other relaxation strategy.

Acknowledgements

The author thanks an anonymous referee whose insightful comments resulted in a substantially improved document.

References

1. Adjya, N., Tawarmalani, M. and Sahinidis, N.V. (1999), A Lagrangian Approach to the Pooling Problem, *Industrial & Engineering Chemistry Research* 38, 1956–1972.

2. Al-Khayyal, F.A., Horst, R. and Pardalos, P.M. (1992), Global Optimization of Concave Functions subject to Quadratic Constraints: An Application in Nonlinear Bilevel Programming, *Annals of Operations Research* 34, 125–147.
3. Barrientos, O. and Correa, R. (2000), An Algorithm for Global Minimization of Linearly Constrained Quadratic Functions, *Journal of Global Optimization* 16, 77–93.
4. Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993), *Nonlinear Programming Theory and Algorithms*, John Wiley & Sons, New York.
5. Ben-Tal, A., Eiger, G. and Gershovitz, V. (1994), Global Minimization by Reducing the Duality Gap, *Mathematical Programming* 63, 193–212.
6. Dur, M. and Horst, R. (1997), Lagrange Duality and Partitioning Techniques in Nonconvex Global Optimization, *Journal of Optimization Theory and Applications* 95, 347–369.
7. Falk, J.E. (1969), Lagrange Multipliers and Nonconvex Programs, *SIAM Journal of Control* 7, 534–545.
8. Floudas, C.A. and Visweswaran, V. (1993), Primal-Relaxed Dual Global Optimization Approach, *Journal of Optimization Theory and Applications* 78, 187–225.
9. Guignard, M. and Rosenwein M.B. (1989), An Application-Oriented Guide for Designing Lagrangean Dual Ascent Algorithms, *European Journal of Operational Research* 43, 197–205.
10. Horst, R. and Raber, U. (1998), Convergent Outer Approximation Algorithms for Solving Unary Problems, *Journal of Global Optimization* 13, 123–149.
11. Kearfott, R.B. (1996), *Rigorous Global Search: Continuous Problems, Nonconvex Optimization and Its Applications* 13, Kluwer Academic Publishers, Dordrecht.
12. Khammash, M.H. (1996), Synthesis of Globally Optimal Controllers for Robust Performance to Unstructured Uncertainty, *IEEE Transactions on Automatic Control* 41, 189–198.
13. Kohl, N. and Madsen, O.B.G. (1997), An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation, *Operations Research* 45, 395–406.
14. Kuno, T. and Utsuomiya, T. (2000), A Lagrangian Based Branch-and-Bound Algorithm for Production-Transportation Problems, *Journal of Global Optimization* 18, 59–73.
15. Li, D. and Sun, X.L. (2000), Local Convexification of the Lagrangian Function in Nonconvex Optimization, *Journal of Optimization Theory and Applications* 104, 109–120.
16. Lodwick, W.A. (1992), Preprocessing Nonlinear Functional Constraints with Applications to the Pooling Problem, *ORSA Journal on Computing* 4, 119–131.
17. Preisig, J.C. (1996), Copositivity and the Minimization of Quadratic Functions with Nonnegativity and Quadratic Equality Constraints, *SIAM Journal on Control and Optimization* 34, 1135–1150.
18. Ryoo, H. S. and Sahinidis, N. V. (1996), A Branch-and-Reduce Approach to Global Optimization, *Journal of Global Optimization* 8, 107–138.
19. Salapaka, M.V., Khammash, M. and Van Voorhis, T. (1998), Synthesis of Globally Optimal Controllers in ℓ_1 using the Reformulation–Linearization Technique, *Proceedings of the IEEE Conference on Decision and Control*, Tampa, FL, December 1998.
20. Sherali, H.D. and Tuncbilek, C.H. (1997), New Reformulation Linearization/Convexification Relaxations for Univariate and Multivariate Polynomial Programming Problems. *Operations Research Letters* 21, 1–9.
21. Sherali, H.D. and Tuncbilek, C.H. (1995), A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems, *Journal of Global Optimization* 7, 1–31.
22. Sherali, H.D. and Tuncbilek, C.H. (1992), A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation–Linearization Technique, *Journal of Global Optimization* 2, 101–112.

23. Thng, I., Cantoni, A. and Leung, Y.H. (1996), Analytical Solutions to the Optimization of a Quadratic Cost Function subject to Linear and Quadratic Equality Constraints, *Applied Mathematics and Optimization* 34, 161–182.
24. Tuy, H. (1997), *Convex Analysis and Global Optimization*, *Nonconvex Optimization and Its Applications* 22, Kluwer Academic Publishers, Dordrecht.
25. Van de Velde, S.L. (1995), Dual Decomposition of a Single-Machine Scheduling Problem, *Mathematical Programming* 69, 413–428.
26. Visweswaran, V. and Floudas, C.A. (1990), A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs-II. Application of Theory and Test Problems, *Computers and Chemical Engineering* 14, 1419–1434.
27. Wah, B.W. and Wang, T. (1999), Efficient and Adaptive Lagrange-Multiplier Methods for Nonlinear Continuous Global Optimization, *Journal of Global Optimization* 14, 1–25.