Journal of Global Optimization **12:** 267–283, 1998.

Global Optimization of Nonconvex Polynomial Programming Problems Having Rational Exponents

HANIF D. SHERALI
Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118, U.S.A.

(Received 28 March 1996; accepted 23 March 1997)

Abstract. This paper considers the solution of nonconvex polynomial programming problems that arise in various engineering design, network distribution, and location-allocation contexts. These problems generally have nonconvex polynomial objective functions and constraints, involving terms of mixed-sign coefficients (as in signomial geometric programs) that have rational exponents on variables. For such problems, we develop an extension of the *Reformulation-Linearization Technique (RLT)* to generate linear programming relaxations that are embedded within a branch-and-bound algorithm. Suitable branching or partitioning strategies are designed for which convergence to a global optimal solution is established. The procedure is illustrated using a numerical example, and several possible extensions and algorithmic enhancements are discussed.

Key words: Polynomial programs, Reformulation-Linearization Technique (RLT), Nonconvex programming, Global optimization.

1. Introduction

In this paper, we consider the global optimization of polynomial programming problems of the following form.

$$\textbf{PP}(\Omega): \quad \text{Minimize} \quad \phi_0(x) \tag{1a}$$

$$\text{subject to} \quad \phi_i(x) \leq \beta_i \text{ for } i = 1, \ldots, m \tag{1b}$$

$$x \in \Omega \equiv \{x : 0 \leq \ell_j \leq x_j \leq u_j < \infty \quad \forall j \in N \equiv \{1, \ldots, n\}\}, \tag{1c}$$

where,

$$\phi_i(x) = \sum_{t \in T_i} \alpha_{it} \prod_{j \in J_{it}} x_j^{\gamma_{itj}} \quad \text{for all } i = 0, 1, \ldots, m. \tag{1d}$$

Here, the objective function as well as each constraint is represented by a polynomial $\phi_i(x)$ that is comprised of terms indexed by a set T_i, where each term $t \in T_i$ has some real coefficient α_{it} that may be of either sign, and is composed of products of monomials $x_j^{\gamma_{itj}}$ for j belonging to some subset J_{it} of N. The indices in J_{it} are assumed to be distinct, each γ_{itj} is assumed to be positive and rational, and $\ell_j < u_j \ \forall j = 1, \ldots, n$. For convenience in exposition, any equality constraint is assumed to be treated here as an equivalent pair of oppositely restricted inequalities.

Note that whenever any negative exponent terms involving some variable x_j are present in a constraint, we can multiply this constraint throughout by x_j^p, where p is the greatest absolute value of the negative exponents appearing for x_j in this constraint, and repeat this for each variable in each constraint to cast the problem into the form (1). (If negative exponent terms appear in the objective function, then this can be handled by rewriting the problem as that of minimizing z, subject to $z \geq \phi_0(x)$ in addition to (1b, 1c).) Alternatively, for any variable x_j that appears in the problem with a negative exponent, we can substitute $y_j = x_j^{-1}$ in each such negative exponent monomial, and include the constraint $x_j y_j = 1$ within the problem in order to obtain an equivalent problem of the form (1). Although the latter approach introduces additional variables in the problem, its advantage is that it avoids the creation of several nonconvex product terms that the former method might generate. However, by the nature of the proposed algorithm, both methods are viable approaches that are open to further investigation.

Problems of the above type find a wide range of applications in production planning, location, and distribution contexts (see Horst and Tuy, 1993) in risk management problems (see Sherali et al., 1994), and in various chemical process design (pooling and blending) and engineering design situations (see Duffin et al. 1967, Dembo 1976, Peterson 1976, Floudas and Pardalos 1990, Floudas and Visweswaran 1990, 1995, Lasdon et al. 1979, and Shor 1990).

In particular, specialized algorithms have been developed to globally optimize Problem PP(Ω) when the exponents γ_{itj} in (1d) are positive *integers*. In this case, Floudas and Visweswaran (1990, 1995) employ a successive quadrification process to convert this problem into an equivalent quadratic polynomial program (see also Shor, 1990), and then develop an extended version of the generalized Benders' algorithm to handle the inherent nonconvexity in the problem. Al-Khayyal et al. (1994) treat this same problem by further converting it to an equivalent bilinearly constrained bilinear program, for which convex envelope based linear programming relaxations are generated that are embedded within a branch-and-bound algorithm. Sherali and Tuncbilek (1992) have developed a Reformulation-Linearization Technique (RTL) for directly generating linear programming relaxations for the polynomial program itself, and have designed globally convergent branch-and-bound algorithms using these relaxations. The resulting relaxations have been shown to theoretically dominate the relaxations that would be obtained by applying RLT to an equivalent quadratic polynomial problem (see Sherali and Tuncbilek, 1997), and moreover, the latter relaxations strictly subsume those of Al-Khayyal et al. (1994) as well.

However, there are several applications in which Problem PP(Ω) arises wherein the variable exponents in (1d) are rational, but non-integral. Examples of such instances include water distribution network design problems (Sherali and Smith, 1995), location-allocation problems using more accurate, empirically determined, ℓ_p distance measures (Brimberg and Love, 1991), several engineering design applications such as the design of heat exchangers and pressure vessels as described in Floudas and Pardalos (1990, Chapters 4, 7, and 11), as well as other design, equilibrium, and economics problems that are typically modeled as geometric programming problems (see Kortanek et al., 1995, and the references cited therein). Although rational exponents can be theoretically intergerized through variable substitutions and suitable transformations, this approach can lead to several additional variables and nonconvex constraints, as well as result in high-degree polynomial terms, thereby exacerbating the complexity of the problem. Moreover, in several of the aforementioned applications, the rational exponents are frequently not amenable to such transformations. For example, in the pipe network design problem solved by Sherali and Smith (1995), the pressure head-loss constraints contain an exponent of 1.852 for the flow variables. It is therefore preferable to derive a method that can directly handle such rational exponent terms.

Note also that Problem PP(Ω) has the form of signomial geometric programming problems (see, for example, Dembo, 1978; Kortanek et al., 1995; and Rickaert and Martens, 1978), except that the lower bounds on the variables are permitted to be zero in PP(Ω), thereby excluding (theoretically) the use of logarithmic/exponential transformations employed in geometric programming where the variables are restricted to be *positive valued*. Cole et al. (1980) consider such problems and transform them into a form that involves sets of posynomial and reversed geometric constraints. An approximating linear program in the logarithm of the variables is then developed by generating tangential hypersurfaces to the posynomial constraints and single-term approximating polynomial constraints to the reversed geometric constraint. The RLT approach we employ also generates linear programming relaxations (in a higher dimensional space that includes the original variables), but via a very different process that attempts to approximate the convex hull of feasible solutions, where the objective function is conceptually accommodated within the constraint set. Cole et al. (1985) have also developed primal and dual cutting plane procedures for the special case of posynomial geometric programming problems. Another successive convex programming branch-and-bound approach has been developed by Passy (1978) for nonconvex problems defined via upper level sets of sums of quasi-concave functions, including signomial geometric programs that can be thus represented. More recently, Maranas and Floudas (1994) have proposed a branch-and-bound algorithm for signomial geometric programs based on using the exponential transformation along with convex envelope approximations for the nonconvex terms in order to generate convex programming approximations. Comparison of the relaxations thus produced with those generated by an RLT approach are presented in Sherali and Tuncbilek (1996), where the latter is empirically demonstrated to yield significantly tighter bounds. Aside from such algorithms developed for geometric programs, other possible approaches for solving PP(Ω) are based on the use of interval arithmetic (see Hansen et al. 1993) or homotopy methods (see Watson et al., 1987, and Kostreva and Kinard, 1991). However, these latter approaches require the determination of all solutions to the

Fritz John necessary optimality conditions in order to recover a global optimum, which can be an arduous task.

The method proposed in this paper to globally solve Problem PP(Ω) is based on an extension of the RLT approach of Sherali and Tuncbilek (1992) to handle rational exponents. Similar to the latter approach, a branch-and-bound algorithm is developed that solves a sequence of relaxations over partitioned subsets of Ω in order to find a global optimum solution. However, to generate the relaxation for each node subproblem, an additional initial step is introduced that constructs an approximating polynomial program having integer exponents to which RLT is subsequently applied. Furthermore, in order to ensure convergence to a global optimum, special partitioning procedures are proposed to coordinate the two levels of relaxations that are involved in this scheme. This gives the basic structure of the algorithm to which several expedients and reduction or bound tightening strategies can be applied as discussed in order to enhance the solution procedure.

The remainder of this paper is organized as follows. Section 2 presents the RLT scheme for generating the relaxations and Section 3 describes the proposed branch-and-bound algorithm in which these relaxations are embedded, and establishes its convergence. The procedure is illustrated in Section 4 using a numerical example, and Section 5 provides a summary along with some possible extensions.

2. A reformulation-linearization technique for generating relaxations

The principal construct in the development of a solution procedure for solving Problem PP(Ω) is the construction of a linear programming relaxation for obtaining lower bounds for this problem, as well as for its partitioned subproblems. For convenience in exposition, let us assume for now that Ω represents either the initial bounds on the variables of the problem, or modified bounds as defined for some partitioned subproblem in a branch-and-bound scheme. The proposed strategy for generating this linear programming relaxation is to apply the Reformulation-Linearization Technique (RLT) of Sherali and Tuncbilek (1992) with an additional initial step that generates an approximating polynomial program having integer variable exponents.

Toward this end, let us define the fractional part of each exponent γ_{itj} by f_{itj}, where,

$$\gamma_{itj} = \lfloor \gamma_{itj} \rfloor + f_{itj}, 0 \leq f_{itj} < 1 \quad \forall i = 0, 1, \ldots, m, \\ t \in T_i, \text{ and } j \in J_{it}. \tag{2}$$

Accordingly, let us denote

$$J_{it}^{+} = \{j \in J_{it} : f_{itj} > 0\} \quad \forall i, t. \tag{3}$$

Furthermore, let $C_\Omega(\cdot)$ denote the convex envelope of $(\cdot)$ over the hyperrectangle Ω (see, for example, Horst and Tuy, 1993). In particular, for each $j \in J_{it}^{+}$, let $C_\Omega(x_j^{f_{itj}})$

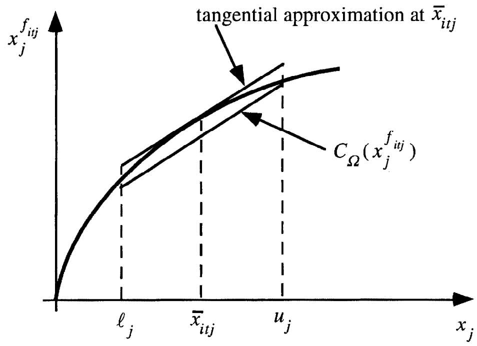

Figure 1. Bounding affine approximations for $x_j^{f_{itj}}$.

represent the (affine) convex envelope of $x_j^{f_{itj}}$ over the interval $\ell_j \leq x_j \leq u_j$. This function is given by (see Figure 1)

$$C_\Omega(x_j^{f_{itj}}) = \ell_j^{f_{itj}} + \frac{(x_j - \ell_j)}{(u_j - \ell_j)} \left[u_j^{f_{itj}} - \ell_j^{f_{itj}} \right]. \tag{4}$$

Additionally, for each $j \in J_{it}^+$, let $\bar{x}_{itj}$ be some point in the (open) interval (ℓ_j, u_j); Lemma 1 later prescribes a selection of this point. Then, by the concavity of $x_j^{f_{itj}}$, we have (see Figure 1),

$$C_\Omega(x_j^{f_{itj}}) \leq x_j^{f_{itj}} \leq \bar{x}_j^{f_{itj}} + [f_{itj}\bar{x}_{itj}^{(f_{itj}-1)}](x_j - \bar{x}_{itj}). \tag{5}$$

Consequently, from (2) and (5), we have,

$$x_j^{\lfloor \gamma_{itj} \rfloor} C_\Omega(x_j^{f_{itj}}) \leq x_j^{\gamma_{itj}} \leq x_j^{\lfloor \gamma_{itj} \rfloor} \left\{ \bar{x}_{itj}^{f_{itj}} + \left[f_{itj}\bar{x}_{itj}^{(f_{itj}-1)} \right] (x_j - \bar{x}_{itj}) \right\}$$
$$\text{for all } j \in J_{it}^+, \quad \forall i, t. \tag{6}$$

Letting $\Omega_j \equiv \{x_j : \ell_j \leq x_j \leq u_j\}$, and denoting $g_{itj}^{\Omega_j}(x_j)$ and $h_{itj}^{\Omega_j}(x_j)$ as, respectively, the lower and upper bounding polynomial functions in (6), each of which have integral exponents, we have,

$$\alpha_{it} \prod_{j \in J_{it}} x_j^{\gamma_{itj}} \geq \phi_{it}^{R(\Omega)}(x) \equiv \begin{cases} \alpha_{it} \prod_{j \in J_{it}} g_{itj}^{\Omega_j}(x_j) & \text{if } \alpha_{it} > 0 \\ \alpha_{it} \prod_{j \in J_{it}} h_{itj}^{\Omega_j}(x_j) & \text{if } \alpha_{it} < 0 \end{cases} \quad \forall i, t, \tag{7}$$

where we have used the fact that $g_{itj}^{\Omega_j}(x_j) = h_{itj}^{\Omega_j}(x_j) = x_j^{\gamma_{itj}}$ whenever γ_{itj} is integral, i.e., $f_{itj} = 0$, or $j \in J_{it} - J_{it}^+$. Hence, summing (7) over all the terms $t \in T_i$ for each $i = 0, 1, \ldots, m$, and denoting the resulting right-hand side $\sum_{t \in T_i} \phi_{it}^{R(\Omega)}(x)$ in this sum as $\phi_i^{R(\Omega)}(x)$, we have,

$$\phi_i(x) \geq \phi_i^{R(\Omega)}(x) \quad \forall x \in \Omega, \quad i = 0, 1, \ldots, m. \tag{8}$$

Accordingly, we construct the corresponding approximation polynomial programming relaxation **PPR**(Ω) as follows.

$$\textbf{PPR}(\Omega):\ \text{Minimize}\{\phi_0^{R(\Omega)}(x):\ \phi_i^{R(\Omega)}(x) \leq \beta_i \quad \forall i = 1, \ldots, m, x \in \Omega\}. \tag{9}$$

REMARK 1. The fact that PPR(Ω) provides a lower bounding relaxation for Problem PP(Ω) follows directly from (8). Moreover, PPR(Ω) has integer exponents for all the variables, and from (6) and (7), we note that the degree δ of this problem, defined as the highest degree of any polynomial term in this problem, is given by the degree of the polynomial program that would be obtained by rounding up all fractional exponents in (1d).

Before proceeding further, let us address the selection of $\bar{x}_{itj}$ in (5) for constructing the approximating polynomial program (9). The following lemma motivates this choice (see Figure 1).

LEMMA 1. *Consider the univariate function $f(x) = x^\alpha$ for $0 \leq \ell \leq x \leq u < \infty$, where $\ell < u$ and $0 < \alpha < 1$. Define $\nabla = (u^\alpha - \ell^\alpha)/(u - \ell)$. Then, the point $\bar{x}$ at which the tangential supporting function for f yields the minimum value of the maximum discrepancy between f and this affine tangential support over $\ell \leq x \leq u$, is given by*

$$\bar{x} = \left(\frac{\alpha}{\nabla}\right)^{1/(1-\alpha)}. \tag{10}$$

Moreover, this point $\bar{x}$ also corresponds to the point of maximum discrepancy between f and the convex envelope of f over $\ell \leq x \leq u$.

Proof. Consider the point $\bar{x}^*$ at which the tangential supporting function yields the same discrepancy Δ, say, at both the endpoints ℓ and u with respect to the given function f. By the concavity of f, since the maximum discrepancy with respect to any tangential support will occur at ℓ or u, and since for $\bar{x} > \bar{x}^*$, the discrepancy at $x = \ell$ will exceed Δ, and similarly, for $\bar{x} < \bar{x}^*$, the discrepancy at $x = u$ will exceed Δ, the desired point of tangential support that would minimize the maximum discrepancy is given by $\bar{x}^*$. At this point $\bar{x} = \bar{x}^*$, we have,

$$[f(\bar{x}) + (\ell - \bar{x})f'(\bar{x})] - f(\ell) = [f(\bar{x}) + (u - \bar{x})f'(\bar{x})] - f(u)$$

i.e., $f'(\bar{x}) = [f(u) - f(\ell)]/(u - \ell) \equiv \nabla$ as defined by the lemma. Hence, $\alpha\bar{x}^{\alpha-1} = \nabla$ or $\bar{x} = (\alpha/\nabla)^{1/(1-\alpha)}$. Moreover, by (4), since the difference between f and its complex envelope over $\ell \leq x \leq u$ is given by $f(x) - [f(\ell) + \nabla(x - \ell)]$, by the concavity of f, this difference is maximized at the point where $f'(x) = \nabla$, i.e., at $\bar{x}$ given by (10). This completes the proof. □

REMARK 2. In accordance with Lemma 1, we select

$$\bar{x}_{itj} = \left[\frac{f_{itj}(u_j - \ell_j)}{(u_j^{f_{itj}} - \ell_j^{f_{itj}})}\right]^{1/(1-f_{itj})} \quad \forall i, t, \text{ and } j \in J_{it}^{+}. \tag{11}$$

We also comment here that while only one point of tangential support has been suggested in our derivation, some special applications might permit the use of more than one tangential approximating constraint. (In general, such a strategy could result in a combinatorial population of constraints.) For example, in the water distribution network design model described in Sherali and Smith (1995), each polynomial constraint involves only a single variable having a rational exponent for which the tangential supporting approximation is necessary in constructing PPR, and in this instance, additional supports can be generated to obtain a tighter relaxation PPR in (9), while maintaining a manageable number of constraints. Furthermore, in the spirit of (5) and (6), various bounding functions can be devised for the monomials $x_j^{\gamma_{itj}}, j \in J_{it}^{+} \quad \forall (i,t)$, and can be suitably coordinated with a partitioning strategy as introduced in the sequel, in order to induce convergence. For example, denoting $C^{\Omega}(\cdot)$ as the concave envelope, over Ω, we can use

$$h_{itj}^{\Omega_j}(x_j) = x_j^{\lfloor \gamma_{itj}-1 \rfloor} C^{\Omega}\left(x^{\{\gamma_{itj} - \lfloor \gamma_{itj}-1 \rfloor\}}\right) \quad \forall j \in J_{it}^{+} \ni \gamma_{itj} > 1, \quad \forall (i,t),$$

as an upper bounding approximation, where C^{Ω} is an affine function in this case. This type of an approximation was used in lieu of (6) in the application described by Sherali and Smith (1995), and yielded comparatively favorable results. In general, one might expect this to be the case when the polynomial expressions in the objective or constraint functions have terms that involve dissimilar variables.

Before proceeding, we establish another property related to the function addressed in Lemma 1 that will be useful in composing partitioning strategies for our branch-and-bound algorithm.

LEMMA 2. *Consider the univariate function* $f(x) = x^{\alpha}$ *for* $0 \leq \ell \leq x \leq u < \infty$, *where* $\ell < u$ *and* $0 < \alpha < 1$. *Suppose that starting with* $[\ell_1, u_1] \equiv [\ell, u]$, *for each* $k \geq 1$, *we determine* $\bar{x}_k$ *via (10) for* $\ell \equiv \ell_k$ *and* $u \equiv u_k$, *and we let* $[\ell_{k+1}, u_{k+1}]$ *be given by either of the partitioned subintervals* $[\ell_k, \bar{x}_k]$ *or* $[\bar{x}_k, u_k]$. *Then, the sequence of intervals* $\{[\ell_k, u_k]\}$ *thus generated tends to* $[\ell^*, u^*]$ *where* $\ell^* = u^*$ *as* $k \to \infty$.

Proof. Since each sequence $\{\ell_k\}$ and $\{u_k\}$ is monotone and bounded, we have that $\{[\ell_k, u_k]\}$ converges to some interval $[\ell^*, u^*]$, where $\ell^* \leq u^*$. Now, on the contrary, assume that $\ell^* < u^*$. Then, by (10), since $\bar{x}_k = [\alpha(u_k - \ell_k)/(u_k^{\alpha} - \ell_k^{\alpha})]^{1/(1-\alpha)}$, we have that $\{\bar{x}_k\} \to \bar{x}^*$ where $\bar{x}^* = [\alpha(u^* - \ell^*)/(u^{*\alpha} - \ell^{*\alpha})]^{1/(1-\alpha)}$. By the strict concavity of f and the Mean Value Theorem, we known that $\bar{x}^* \in (\ell^*, u^*)$ while $\bar{x}_k \notin (\ell_{k'}, u_{k'})\ \forall k' > k$. This means that we must have $\bar{x}^* = \ell^*$ or u^*, a contradiction. This completes the proof. □

We now present the proposed RLT scheme for generating linear programming relaxations for Problem PP(Ω). This scheme operates in two phases. In the first phase, which is the *Reformulation phase*, the relaxed polynomial program PPR(Ω) is constructed, and additional implied inequalities are generated and added to this problem. The second phase is known as the *Linearization phase*, in which the resulting problem is transformed into a linear program by substituting a single variable for each distinct variable-product term. Specific details of these two phases are described below.

Reformulation phase

Step I: Given PP(Ω), generate the relaxed polynomial program PPR(Ω) given by (9) as described above. Let δ denote the degree of PPR(Ω) (see Remark 1).

Step II: Let $\bar{N} = \{N, \ldots, N\}$ denote δ replicates of N. Compose all possible distinct constraints of the type

$$F_\delta(J_1, J_2) \equiv \prod_{j \in J_1} (x_j - \ell_j) \prod_{j \in J_2} (u_j - x_j) \geq 0 \quad \forall J_1 \cup J_2 \subseteq \bar{N}, |J_1 \cup J_2| = \delta, \tag{12}$$

obtained by taking products of the *bounding factors* $(x_j - \ell_j) \geq 0$ and $(u_j - x_j) \geq 0$ δ at a time, including possible repetitions. (Here, $J_1 \cup J_2$ denotes the joint collection of indices within J_1 and J_2, *preserving repetitions*.) Augment Problem PPR(Ω) by adding these constraints (12) to it. (In addition, *optionally*, other implied polynomial constraints of degree less than or equal to δ can be generated by taking suitable inter-products involving the joint collection of *constraint factors* $\beta_i - \phi_i^{R(\Omega)}(x) \geq 0, i = 1, \ldots, m$, and the aforementioned bound factors, and these can be added to PPR(Ω) in order to further tighten the relaxation obtained via this overall RLT process.)

Linearization phase

Linearize the resulting polynomial program obtained at the end of the Reformulation Phase by substituting

$$X_J = \prod_{j \in J} x_j \quad \forall J \subseteq \bar{N}, \tag{13}$$

where the indices in J are assumed to be sequenced in nondecreasing order, and where $X_{\{j\}} \equiv x_j \; \forall j \in N$, and $X_\phi \equiv 1$. Denote the resulting linear program thus produced by **LP**(Ω).

The following results establish some salient properties of Problem LP(Ω) that are essential in designing the proposed algorithm. Notationally, for any Problem P, let us denote the optimal objective function value of P by ν(P).

LEMMA 3. *$\nu[LP(\Omega)] \leq \nu[PP(\Omega)]$, and so, $LP(\Omega)$ provides a lower bound on the polynomial program $PP(\Omega)$.*

Proof. Obvious by construction. □

LEMMA 4. *Let* $(\hat{x}, \hat{X})$ *be any feasible solution to LP*(Ω). *Suppose that* $\hat{x}_p = \ell_p$ *for some* $p \in N$. *Then* $\hat{X}_{J \cup p} = \ell_p \hat{X}_J \, \forall J \subseteq \bar{N}, 1 \leq |J| \leq \delta - 1$. *Similarly,* $\hat{x}_p = u_p$ *implies that* $\hat{X}_{J \cup p} = u_p \hat{X}_J \; \forall J \subseteq \bar{N}, 1 \leq |J| \leq \delta - 1$.

Proof. See Sherali and Tuncbilek (1992). □

3. A branch-and-bound algorithm

The proposed algorithm is a branch-and-bound approach that is based on partitioning the set Ω into sub-hyperrectangles, each associated with a node of the branch-and-bound tree. Hence, at any stage s of the algorithm, suppose that we have a collection of active nodes indexed by $q \in Q_s$, say, each associated with a hyperrectangle $\Omega^q \subseteq \Omega$, $\forall q \in Q_s$. For each such node, we will have computed a lower bound LB_q via the solution of the linear program LP(Ω^q) (see Lemma 3), so that the lower bound on PP(Ω) at stage s is given by $LB(s)$ = minimum $\{LB_q : q \in Q_s\}$. Whenever the lower bounding solution (or some perturbation thereof) for any node subproblem turns out to be feasible to PP(Ω), we update the upper bound or incumbent solution value ν^*, if necessary. Hence, the active nodes all satisfy $LB_q < \nu^* \; \forall q \in Q_s$, for each stage s. We now select an active node $q(s)$ that yields the least lower bound $LB(s) \equiv LB_{q(s)}$ among $q \in Q_s$, and we partition its associated hyperrectangle into two sub-hyperrectangles as described below, computing the lower bounds for each new node as before. Upon fathoming any nonimproving nodes, we obtain a collection of active nodes for the next stage, and this process is repeated until convergence is obtained.

The critical element in guaranteeing convergence to a global minimum is the choice of a suitable partitioning strategy. Three such branching rules that assure convergence for the proposed algorithm are stated below. The first of these (Rule (A)) is a simple, standard bisection rule. While this is sufficient to ensure convergence since it drives all the intervals to zero for the variables that are associated with the term that yields the greatest discrepancy in the employed approximation along any infinite branch of the branch-and-bound tree, it is not too cognizant of the nature and solution of the employed relaxations. Rule (B) below accordingly suggests branching at the value $\bar{x}_{itj}$ given by (11) whenever the approximating problem employs the tangential approximation at this point for the term under consideration, motivated by Lemmas 1 and 2 and the fact that the tangential approximation is exact at this point. Likewise, Rule (C) is further oriented toward computational effectiveness by combining Rule (B) with a partitioning at the linear programming relaxation value whenever admissible, since as motivated by Lemma 4, the RLT approximation involving any variable becomes exact (in the sense stated in the Lemma) whenever this variable coincides with one of its bounds. Hence, by letting the current linear programing relaxation value become an upper interval bound on one partitioned subnode and a lower interval bound on the other, we encourage the tightening of the resulting relaxations. Further motivation is provided with the detailed statement given below.

Branching rules
Consider any node subproblem identified by the hyperrectangle $\Omega' \subseteq \Omega$, and let $(\hat{x}, \hat{X})$ represent the solution obtained for its associated linear programming relaxation LP(Ω'). Determine the term (r, τ) in the polynomial program for which the discrepancy between its value computed at $\hat{x}$ and the value of its lower bounding function $\phi_{r\tau}^{R(\Omega')}(x)$ given by (7) and linearized under (13) in LP(Ω'), computed at $(\hat{x}, \hat{X})$, is a maximum. Letting $\phi_{r\tau}^{L}(\hat{x}, \hat{X})$ denote the latter linearized value, we have that

$$\alpha_{r\tau} \prod_{j \in J_{r\tau}} \hat{x}_j^{\gamma_{r\tau j}} - \phi_{r\tau}^{L(\Omega')}(\hat{x}, \hat{X}) = \underset{i=0,\ldots,m, t \in T_i}{\text{maximum}} \left\{ \alpha_{it} \prod_{j \in J_{it}} \hat{x}_j^{\gamma_{itj}} - \phi_{it}^{L(\Omega')}(\hat{x}, \hat{X}) \right\}. \tag{14}$$

The selection of the branching variable x_p and the partitioning of Ω' is then done using one of the following rules, where $\Omega' \equiv \{x : \ell'_j \leq x_j \leq u'_j \ \forall j \in N\}$.

Rule (A). Let $p = \text{argmax}\{u'_j - \ell'_j : j \in J_{r\tau}\}$, and partition Ω' by bisecting the interval $[\ell'_p, u'_p]$ into the subintervals $[\ell'_p, (\ell'_p + u'_p)/2]$ and $[(\ell'_p + u'_p)/2, u'_p]$.

Rule (B). Let $p = \text{argmax}\{u'_j - \ell'_j : j \in J_{r\tau}\}$. Motivated by Lemma 1, partition Ω' by subdividing the interval $[\ell'_p, u'_p]$ into $[\ell'_p, \bar{x}_{r\tau p}]$ and $[\bar{x}_{r\tau p}, u'_p]$ if $f_{r\tau p} > 0$, where $\bar{x}_{r\tau p}$ is given by (11) for the current bounds on x_p, and by bisecting $[\ell'_p, u'_p]$ if $f_{r\tau p} = 0$. Alternatively, motivated by (7), this rule can be modified so that it is applied when $\alpha_{r\tau} < 0$, with the bisection rule being used if $\alpha_{r\tau} > 0$.)

Rule (C). For each $j \in J_{r\tau}$ compute

$$\begin{aligned} &\theta_j = \min\{\hat{x}_j - \ell'_j, u'_j - \hat{x}_j\} \text{ and let } \tilde{x}_j = \hat{x}_j, \text{ if } \alpha_{r\tau} > 0 \\ &\quad \text{or if } \alpha_{r\tau} < 0 \text{ and } f_{r\tau j} = 0, \end{aligned} \tag{15a}$$

and compute

$$\begin{aligned} &\theta_j = \max\{|\hat{x}_j - \bar{x}_{r\tau j}|, \ \min\{\hat{x}_j - \ell'_j, u'_j - \hat{x}_j\}\} \\ &\quad \text{and let } \tilde{x}_j = \bar{x}_{r\tau j}, \text{ otherwise}. \end{aligned} \tag{15b}$$

Note that in the case of (15a), we have used the left inequality in (6) for the approximation (7), and this inequality holds as an equality in case $\hat{x}_j$ equals ℓ'_j or u'_j, and moreover, by Lemma 4, $\hat{x}_j \hat{X}_J \equiv \hat{X}_{J \cup j} \ \forall J$ in this case. On the other hand, for the case of (15b) where $\alpha_{r\tau} < 0$ and $f_{r\tau j} > 0$, we have used the right inequality in (6) for the approximation (7), and we would like $\hat{x}_j$ to coincide with $\bar{x}_{r\tau j}$ to make this hold as an equality, but as in lemma 4, we would also like $\hat{x}_j$ to be close to ℓ'_j or u'_j. Hence, with this motivation of θ_j, in conjunction with our desire of reducing bounding interval lengths, we select $p = \text{argmax}\{(u'_j - \ell'_j)\theta_j : j \in J_{r\tau}\}$, and we partition Ω' by subdividing the interval $[\ell'_p, u'_p]$ into $[\ell'_p, \tilde{x}_p]$ and $[\tilde{x}_p, u'_p]$.

Algorithmic statement

Step 0: Initialization. In the notation used above, initialize by setting $x^* = \emptyset, \nu^* = \infty, s = 1, Q_s = \{1\}, q(s) = 1$, and $\Omega^1 = \Omega$. Solve LP(Ω^1) and let $(\hat{x}, \hat{X})$ be the solution obtained of objective value $LB_1 = \nu[\text{LP}(\Omega^1)]$. If $\hat{x}$ is feasible to PP(Ω^1) (perhaps after using some heuristic perturbation or some Newton–Raphson iterations) update x^* and ν^*, if necessary. If $\nu^* \leq LB_1 + \varepsilon$, where $\varepsilon \geq 0$ is some accuracy tolerance, then stop with x^* as the prescribed solution to Problem PP(Ω^1). Otherwise, select a branching variable x_p according to any one of the above branching rules (A), (B), or (C) and proceed to Step 1.

Step 1: Partitioning step. Partition $\Omega^{q(s)}$ into two sub-hyperrectangles by splitting the interval for x_p according to the selected branching rule. Replace $q(s)$ by these two new node indices in Q_s.

Step 2: Bounding step. Solve the RLT linear programming relaxation for each of the two new nodes generated, and update the incumbent solution if possible, as in the Initialization Step.

Step 3: Fathoming step. Fathom any nonimproving nodes by setting $Q_{s+1} = Q_s - \{q \in Q_s : LB_q + \varepsilon \geq \nu^*\}$. If $Q_{s+1} = \emptyset$ then stop. Otherwise, increment s by one and proceed to Step 4.

Step 4: Node selection step. Select an active node $q(s) \in \text{argmin}\ \{LB_q : q \in Q_s\}$, and return to Step 1.

THEOREM 1 (Convergence result). *The above algorithm (run with $\epsilon \equiv 0$) either terminates finitely with the incumbent solution being optimal to PP(Ω), or else an infinite sequence of stages is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the x-variable part of the linear programming relaxation solutions generated for the node subproblems solves PP(Ω).*

Proof. The case of finite termination is clear. Hence, suppose that an infinite sequence of stages is generated. Consider any infinite branch of the branch-and-bound tree associated with a nested sequence of partitions $\{\Omega^{q(s)}\}$ for stages s in some index set S. Hence,

$$\begin{aligned}\nu[\text{PP}(\Omega)] &\geq LB(s) = LB_{q(s)} \equiv \nu[\text{LP}(\Omega^{q(s)})] \\ &\equiv \sum_{t \in T_0} \phi_{0t}^{L(\Omega^{q(s)})}(x^{q(s)}, X^{q(s)}) \quad \forall s \in S,\end{aligned} \tag{16}$$

where, for each node $q(s), s \in S, (x^{q(s)}, X^{q(s)})$ denotes the optimum solution obtained for LP($\Omega^{q(s)}$). Moreover, let $\ell^{q(s)}$, and $u^{q(s)}$ be the associated vectors of lower and upper bounds that define $\Omega^{q(s)}$. By taking any convergent subsequence if necessary, suppose that $\{x^{q(s)}, X^{q(s)}, \ell^{q(s)}, u^{q(s)}\}_S \to (x^*, X^*, \ell^*, u^*)$, and denote $\Omega^* = \{x : \ell^* \le x \le u^*\}$. We must show that x^* then solves Problem PP(Ω).

Now, over the infinite sequence of nodes $\{q(s), s \in S\}$, there exists a term (r, τ) for $\tau \in T_r, r \in \{0, 1, \ldots, m\}$, that is picked infinitely often via (14). Let $S_1 \subseteq S$ be the stages for which a partitioning is done based on this term (r, τ) using Branching Rule (A), (B), or (C). Hence, we have from (14) that,

$$\alpha_{r\tau} \prod_{j \in J_{rt}} [x_j^{q(s)}]^{\gamma_{r\tau j}} - \phi_{r\tau}^{L(\Omega^{q(s)})}(x^{q(s)}, X^{q(s)})$$
$$\ge \alpha_{it} \prod_{j \in J_{it}} [x_j^{q(s)}]^{\gamma_{itj}} - \phi_{it}^{L(\Omega^{q(s)})}(x^{q(s)}, X^{q(s)}) \tag{17}$$
$$\forall i = 0, 1, \ldots, m, t \in T_i, \text{ for each } s \in S_1.$$

Let us consider each branching rule in turn.

Rule A: Under this rule, since the largest interval for $x_j, j \in J_{r\tau}$, is bisected at each node $q(s), s \in S_1$, we have that $l_j^* = u_j^* \quad \forall j \in J_{r\tau}$, and so, $x_j^* = \ell_j^* = u_j^* \quad \forall j \in J_{r\tau}$. But by (7) and Lemma 4, this means that

$$\alpha_{r\tau} \prod_{j \in J_{r\tau}} (x_j^*)^{\gamma_{r\tau j}} = \phi_{r\tau}^{R(\Omega^*)}(x^*) = \phi_{r\tau}^{L(\Omega^*)}(x^*, X^*), \tag{18a}$$

where

$$\phi_{r\tau}^{R(\Omega^*)}(x^*) \equiv \lim_{s \to \infty, s \in S_1} \phi_{r\tau}^{R(\Omega^{q(s)})}(x^{q(s)}) \text{ and}$$
$$\phi_{r\tau}^{L(\Omega^*)}(x^*, X^*) \equiv \lim_{s \to \infty, s \in S_1} \phi_{r\tau}^{L(\Omega^{q(s)})}(x^{q(s)}, X^{q(s)}). \tag{18b}$$

Hence, the left-hand side in (17) approaches zero as $s \to \infty, s \in S_1$, and so, the right-hand side in (17) is nonpositive in the limit. But this means that by taking limits in (17) as $s \to \infty, s \in S_1$, and by the feasibility of $(x^{q(s)}, X^{q(s)})$ for LP($\Omega^{q(s)}$), we have, $\ell \le l^* \le x^* \le u^* \le u$ and

$$\beta_i \ge \sum_{t \in T_i} \phi_{it}^{L(\Omega^*)}(x^*, X^*) \ge \sum_{i \in T_i} \alpha_{it} \prod_{j \in J_{it}} (x_j^*)^{\gamma_{itj}} \quad \forall i = 1, \ldots, m, \tag{19a}$$

where as before,

$$\phi_{it}^{L(\Omega^*)}(x^*, X^*) \equiv \lim_{s \to \infty, s \in S_1} \phi_{it}^{L(\Omega^{q(s)})}(x^{q(s)}, X^{q(s)}) \quad \forall (i, t). \tag{19b}$$

Hence, x^* is feasible to PP(Ω). Moreover, by (16) and (17) for $i = 0$, taking limits as $S \to \infty, s \in S_1$, we get

$$\nu[\text{PP}(\Omega)] \ge \sum_{t \in T_0} \phi_{0t}^{L(\Omega^*)}(x^*, X^*) \ge \sum_{t \in T_0} \alpha_{0t} \prod_{j \in J_{0t}} (x_j^*)^{\gamma_{0tj}} \equiv \phi_0(x^*). \tag{20}$$

Consequently, $\phi_0(x^*)$ must equal ν[PP(Ω)], and hence x^* solves PP(Ω).

Rule B: To establish convergence under this rule, using the argument for Rule A, it is sufficient to show that $l_j^* = u_j^* \ \forall j \in J_{r\tau}$. Since at each step, the largest interval for $j \in J_{r\tau}$ is selected for partitioning, and each such interval is either bisected or is partitioned at the corresponding $\bar{x}_{r\tau j}$ value, by Lemma 2, it readily follows that $l_j^* = u_j^* \ \forall j \in J_{r\tau}$. Hence again, the proof holds true in this case.

Rule C: Following the proof for Rule A, it is sufficient to show that the discrepancy in the term (r, τ) approaches zero as $s \to \infty, s \in S_1$, i.e., Equation (18) holds true. By the partitioning strategy, over the nested sequence of nodes $\{q(s), s \in S_1\}$, there exists some index $p \in J_{r\tau}$ that is selected infinitely often for partitioning according to

$$\theta_p(u'_p - \ell'_p) \geq \theta_j(u'_j - \ell'_j) \quad \forall j \in J_{r\tau}. \tag{21}$$

Let $S_2 \subseteq S_1$ index the set of nodes where this occurs, and let us consider two cases.

Case (i): θ_p is given by (15a). In this case, by (15a) and Rule C, since for each $s \in S_2, \tilde{x}_p \equiv x_p^{q(s)} \notin (\ell_p^{q(s')}, u_p^{q(s')}) \ \forall s' \in S_2, s' > s$, while $x_p^* \in [\ell_p^*, u_p^*]$, we must have that $x_p^* = \ell_p^*$ or $x_p^* = u_p^*$. Hence, the sequence of values $\{\theta_p\} \to 0$ as $s \to \infty, s \in S_2$, and so by (21), either $\{u'_j - \ell'_j\} \to 0$ or $\{\theta_j\} \to 0 \ \forall j \in J_{r\tau}$. But this means that if $j \in J_{r\tau}$ is of the case (15a), then as above, $x_j^* = \ell_j^*$ or $x_j^* = u_j^*$, and if it is of the case (15b), then we again have $x_j^* = \ell_j^* = u_j^*$. Consequently, (7) holds as an equality in the limit, and by Lemma 4, we also have $\phi_{r\tau}^{R(\Omega^*)}(x^*) = \phi_{r\tau}^{L(\Omega^*)}(x^*, X^*)$, and so, (18) holds true.

Case (ii): θ_p is given by (15b). In this case, since for each $s \in S_2$, the point $\tilde{x}_p$ is selected as the corresponding point $\bar{x}_{r\tau p}$, we have as in Lemma 2 that $x_p^* = \ell_p^* = u_p^*$ in the limit as $s \to \infty, s \in S_2$. Hence, the sequence $\{\theta_p\} \to 0$, and so by (21), we again have that $\{(u'_j - \ell'_j)\theta_j\} \to 0 \ \forall j \in J_{t\tau}$ as $s \to \infty, s \in S_2$. By the argument for Case (i), Equation (18) holds true, and this completes the proof. □

4. Illustrative example

Consider the following example adapted from the small, yet notoriously challenging, bilinear programming problem given in Al-Khayyal and Falk (1983).

$$\text{Minimize}\{-x_1 + x_1 x_2^{0.5} - x_2 : -6x_1 + 8x_2 \leq 3, 3x_1 - x_2 \leq 3,$$
$$(0, 0) \leq (x_1, x_2) \leq (1.5, 1.5)\}.$$

Denoting any bounding intervals on the variables x_1 and x_2 by $[\ell_i, u_i]$, for $i = 1, 2$, respectively, we have from (4) and (6) that

$$x_2^{0.5} \geq \ell_2^{0.5} + (x_2 - \ell_2)(u_2^{0.5} - \ell_2^{0.5})/(u_2 - \ell_2).$$

Hence, defining Ω as in (1c), we have from (7) and (8) that the approximating polynomial program PPR(Ω) defined in (9) is given as follows.

PPR(Ω): Minimize$\{(\lambda - 1)x_1 + \mu x_1 x_2 - x_2 : -6x_1 + 8x_2 \leq 3,$

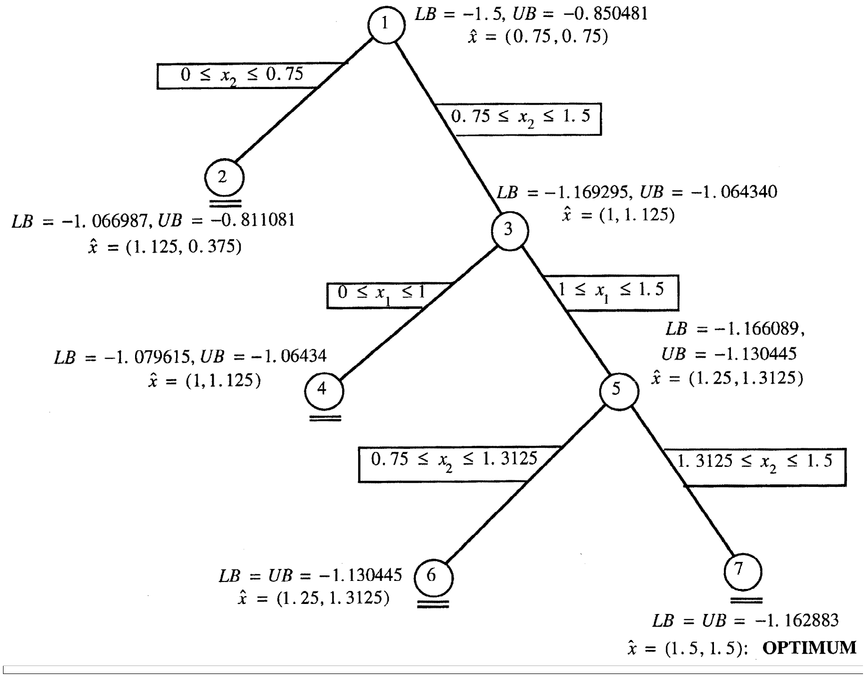

Figure 2. Branch-and-Bound Tree for Branching Rule (C).

$$3x_1 - x_2 \leq 3, x \in \Omega\},$$

where $\lambda \equiv [u_2\ell_2^{0.5} - \ell_2 u_2^{0.5}]/(u_2 - \ell_2)$ and $\mu = [u_2^{0.5} - \ell_2^{0.5}]/(u_2 - \ell_2)$. The degree of this polynomial program is $\delta = 2$. Hence, at Step II of the Reformulation Phase of RLT, we generate all pairwise products (21 in number) of the constraint and bound factors $(3 + 6x_1 - 8x_2) \geq 0, (3 - 3x_1 + x_2) \geq 0$, $(x_1 - \ell_1) \geq 0$, $(u_1 - x_1) \geq 0$, $(x_2 - \ell_2) \geq 0$, and $(u_2 - x_2) \geq 0$, including self-products. These product constraints are linearized by substituting

$$X_{11} = x_1^2, X_{22} = x_2^2, \text{ and } X_{12} = x_1 x_2.$$

For example, the product constraint $(3 + 6x_1 - 8x_2)(3 - 3x_1 + x_2) \geq 0$ yields

$$9 + 9x_1 - 21x_2 - 18X_{11} - 8X_{22} + 30X_{12} \geq 0.$$

The RLT lower bounding linear program LP(Ω) is then to minimize $\{(\lambda - 1)x_1 + \mu X_{12} - x_2\}$ subject to the 21 RLT linearized product constraints, where the original as well as the bounding restrictions on all the variables are easily verified to be implied by these constraints.

Using Branching Rule C, the problem is solved to optimality after enumerating 7 nodes as shown in Figure 2. Note that in this example, the most (actually, only) discrepant term via (14) is the nonlinear term that appears in the objective function, and the interval partitioning is performed at the linear programming solution value as determined via (15a).

As a point of interest, using either the bisection Rule (A) or the Branching Rule (B), the branch-and-bound algorithm enumerated 9 nodes in order to solve the problem.

5. Summary and extensions

In this paper, we have presented a global optimization approach for solving polynomial programming problems that may, in general, have rational exponents, mixed-sign coefficients, and permit variables to take on zero values. As discussed, such problems arise in various location-allocation, chemical process and engineering design, and economic equilibrium problems. For such problems, we have developed a lower bounding scheme that employs two levels of approximation – one that constructs a lower bounding polynomial program having integral exponents, and a second based on the RLT approach. This lower bound is embedded within a branch-and-bound algorithm, and three particular partitioning strategies are developed, each of which simultaneously drives the error in these two levels of approximation to zero, hence inducing convergence to a global optimum. While the proposed method remains to be computationally refined and tested (several enhancements are recommended below), a prototype of this basic approach has been specialized and applied to the pipe network design problem in Sherali and Smith (1995). Using this methodology, a standard test case from the literature, and several of its variants, have been solved for the first time to provable global optimality.

As alluded above, there are several extensions that can be explored to enhance the proposed basic algorithm. First, several other classes of constraints could be developed to further tighten the lower bounding linear program. These can include multiple polynomial approximations for the objective function (and perhaps other key constraints), alternative bounding polynomial functions, as well as various other valid RLT product constraints. Second, a Lagrangian dual approach could be used to cope with and to exploit the size and structure of the relaxations generated. Third, several "preprocessing" or "bound-reduction" strategies can be devised to further tighten the relaxation as in, for example, Shectman and Sahinidis (1995) and Sherali and Tuncbilek (1995). Fourth, as illustrated by our analysis, many alternative branching strategies exist that admit convergence to a global optimum and these need to be investigated and computationally tested. Fifth, suitable heuristics could be designed to determine good quality feasible solutions based on the LP relaxations being solved, in order to enhance the branch-and-bound algorithm, as well as to provide a practical tool for deriving useful solutions to relatively large-scale problems. Finally, as noted by Hansen and Jaumard (1992), polynomial programs can be used to approximate various problems that include trigonometric and transcendental functions. Hence, our methodology can be potentially extended to handle such problems as well. The use of such approximations can also be coordinated with various transformations, such as the exponential/logarithmic transformation used for geometric programs. These investigations are being pursued in ongoing research and further results and computational experience will be forthcoming.

Acknowledgements

This material is based upon research supported by the *National Science Foundation* under Grant Number DMI-9521398, and the Air Force Office of Scientific Research under Grant Number F49620-96-1-0274.

References

Al-Khayyal, F.A. and J.E. Falk (1983), Jointly constrained biconvex programming, *Math. of Oper. Res.* **8**, 273–283.

Al-Khayyal, F.A., C. Larson and T. Van Voorhis (1994), A relaxation method for nonconvex quadratically constrained quadratic programs.

Brimberg, J. and R.F. Love (1991), Estimating travel distances by the weighted ℓ_p norm, *Naval Research Logistics* **38** 241–259.

Cole, F., W. Gochet and Y. Smeers (1985), A comparison between a primal and a dual cutting plane algorithm for posynomial geometric programming problems, *Journal of Optimization Theory and Applications* **47**, 159–180.

Cole, F., W. Gochet, F. Van Assche, J. Ecker and Y. Smeers (1980), Reversed geometric programming: A branch-and-bound method involving linear subproblems, *European Journal of Operational Research* **5**, 26–35.

Dembo, R.S. (1976), A set of geometric programming test problems and their solutions, *Mathematical Programming* **10**, 192–213.

Dembo, R.S. (1978), Current state of the art of algorithms and computer software for geometric programming, *Journal of Optimization theory and Applications* **26**, 149–183.

Duffin, R.J., E.L. Peterson and C. Zener (1967), *Geometric Programming*. John Wiley & Sons, New York.

Floudas, C.A. and P.M. Pardalos (1990), A collection of test problems for constrained global optimization algorithms, *Lecture Notes in Computer Science*, Vol. 455, (eds. G Goos and J. Hartmanis). Springer Verlag, Berlin.

Floudas, C.A. and V. Visweswaran (1995), Quadratic optimization, in *Handbook of Global Optimization*, Nonconvex Optimization and its Applications (eds. R. Horst and P.M. Pardalos). Kluwer Academic Publishers, 217–270.

Floudas, C.A. and V. Visweswaran (1990), A global optimization algorithm (GOP) for certain classes of nonconvex NLP's I: Theory, *Computers and Chemical Engineering* **14**, 1397–1417.

Hansen, P., and B. Jaumard (1992), Reduction of indefinite quadratic programs to bilinear programs, *Journal of Global Optimization* **2**(1), 41–60.

Hansen, P., B. Jaumard and J. Xiong (1993), Decomposition and interval arithmetic applied to global minimization of polynomial and rational functions, *Journal of Global Optimization* **3**, 421–437.

Horst, R. and H. Tuy (1993), *Global Optimization: Deterministic Approaches*, 2nd ed. Springer Verlag, Berlin.

Kortanek, K.L., X. Xu and Y. Ye (1995), An infeasible interior-point algorithm for solving primal and dual geometric programs. Manuscript, Department of Management Science, The University of Iowa, Iowa City, IA 52242.

Kostreva, M.M. and L.A. Kinard (1991), A differentiable homotopy approach for solving polynomial optimization problems and noncooperative games, *Computers Math. Applic.* **21**(6/7), 135–143.

Lasdon, L.S., A.D. Waren, S. Sarkar and F. Palacios, (1979), Solving the pooling problem using generalized reduced gradient and successive linear programming algorithms, *SIGMAP Bull.* **77**, 9–15.

Maranas, C.D. and C.A. Floudas (1994), Global optimization in generalized geometric programming. Working Paper, Department of Chemical Engineering, Princeton University, Princeton, NJ.

Passy, U. (1978), Global solutions of mathematical programs with intrinsically concave functions, *Journal of Optimization Theory and Applications* **26**, 97–115.

Peterson, E.L. (1976), Geometric programming, *SIAM Review* **18**, 1–15.

Rickaert, M.J. and X.M. Martens (1978), Comparison of generalized geometric programming algorithms, *Journal of Optimization Theory and Applications* **26**, 205–242.

Shectman, H.P. and N.V. Sahindis (1994), A finite algorithm for global minimization of separable concave programs. Technical Report, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champagne, IL.

Sherali, H.D., A. Alameddine and T.S. Glickman (1994/95), Biconvex models and algorithms for risk management problems, *American Journal of Mathematical and Management Sciences* **14**(2&3), 197–228.

Sherali, H.D. and E.P. Smith (1995), A global optimization approach to a water distribution network design problem. Research Report #HDS95-6, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA (to appear in the *Journal of Global Optimization*).

Sherali, H.D. and C.H. Tuncbilek (1992), A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique, *Journal of Global Optimization* **2**, 101–112.

Sherali, H.D. and C.H. Tuncbilek (1995), A reformulation-convexification approach for solving nonconvex quadratic programming problems, *Journal of Global Optimization* **7**, 1–31.

Sherali, H.D. and C.H. Tuncbilek (1997), Comparison of two Reformulation-Linearization Technique based linear programming relaxations for polynomial programming problems, *Journal of Global Optimization* **10**, 381–390.

Sherali, H.D. and C.H. Tuncbilek (1996), New reformulation-linearization/convexification relaxations for univariate and multivariate polynomial programming problems, under revision for *Operations Research Letters*.

Shor, N.Z. (1990), Dual quadratic estimates in polynomial and boolean programming, *Annals of Operations Research* **25**, 163–168.

Watson, L.T., S.C. Billups and A.P. Morgan (1987), Algorithm 652 HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Transactions on Mathematical Software* **13**(3), 281–310.