A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation-Linearization Technique

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

(Received: 5 July 1991; accepted: 8 October 1991)

Abstract. This paper is concerned with the development of an algorithm to solve continuous polynomial programming problems for which the objective function and the constraints are specified polynomials. A linear programming relaxation is derived for the problem based on a Reformulation Linearization Technique (RLT), which generates nonlinear (polynomial) implied constraints to be included in the original problem, and subsequently linearizes the resulting problem by defining new variables, one for each distinct polynomial term. This construct is then used to obtain lower bounds in the context of a proposed branch and bound scheme, which is proven to converge to a global optimal solution. A numerical example is presented to illustrate the proposed algorithm.

Key words. Reformulation, linearization, polynomial programs, multilinear programs.

1. Introduction

In this paper, we address a polynomial programming problem which seeks a global minimum to a polynomial objective function subject to a set of polynomial constraint functions, all defined in terms of some continuous decision variables. We do not put any other convexity or generalized convexity restrictions on these functions, but we do assume that the feasible region is compact. A mathematical formulation of this problem is given below.

$$\text{PP}(\Omega): \text{Minimize}\{\phi_0(x): x \in Z \cap \Omega\}\ ,$$

where $Z = \{x: \phi_r(x) \geq \beta_r \text{ for } r = 1, \ldots, R_1,\ \phi_r(x) = \beta_r \text{ for } r = R_1 + 1, \ldots, R\}$, and $\Omega = \{x: 0 \leq l_j \leq x_j \leq u_j < \infty, \text{ for } j = 1, \ldots, n\}$, and where

$$\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt}\Big[\prod_{j \in J_{rt}} x_j\Big] \quad \text{for } r = 0, 1, \ldots, R\ . \tag{1}$$

Here, T_r is an index set for the terms defining $\phi_r(\cdot)$, and α_{rt} are real coefficients for the polynomial terms $(\Pi_{j \in J_{rt}} x_j)$, $t \in T_r$, $r = 0, 1, \ldots, R$. Note that we permit a repetition of indices within each set J_{rt}. For example, if $J_{rt} = \{1, 2, 2, 3\}$, then the corresponding polynomial term is $x_1 x_2^2 x_3$. In particular, let us denote $N = \{1, \ldots, n\}$, and let δ be the maximum degree of any polynomial term appearing in PP(Ω). Define $\bar{N} = \{N, \ldots, N\}$ to be composed of δ replicates of N. Then, each $J_{rt} \subseteq \bar{N}$, with $1 \leq |J_{rt}| \leq \delta$, for $t \in T_r$, $r = 0, 1, \ldots, R$.

Journal of Global Optimization **2**: 101–112, 1992.

In its general form as formulated above, polynomial programs have not received much attention. Notable exceptions are the recent papers by Floudas and Visweswaran (1991) and Shor (1990). The first of these papers suggests a successive quadratic variable substitution strategy to transform a given polynomial problem to one of minimizing a bilinear function subject to bilinear constraints. Following this, a generalized Benders type of approach is employed, where the Benders cuts are replaced by suitable implied linearized Lagrange functions. A branch and bound algorithm is developed to obtain an ε-optimal solution. The second paper adopts similar successive quadratic transformations to write the problem as an equivalent quadratically constrained quadratic problem. In some problem instances, products of linear constraints, if present, are selectively used to generate additional implied quadratic constraints. However, rather than solve such problems in general, the focus here is to develop sufficient conditions under which the Lagrangian dual of the resulting equivalent quadratic problem would preclude the presence of a duality gap.

In contrast, a substantial amount of literature exists on some special cases of Problem PP(Ω), such as concave minimization problems and bilinear programming problems (see Pardalos and Rosen (1987) for a recent survey). Another related problem to PP(Ω) is the 0–1 polynomial program, where the x-variables are restricted to be binary valued. Various linearization, algebraic, enumerative, and cutting plane methods have been developed for nonlinear 0–1 programs, as recently surveyed by Hansen, Jaumard, and Mathon (1989). In particular, for constrained polynomial 0–1 programming problems, the linearization-cutting plane method proposed by Balas and Mazzola (1984a, b) appears to perform quite favorably (see Hansen *et al.*, 1989). However, this approach exploits the 0–1 structure of the problem, and is therefore not directly applicable for continuous polynomial programming problems.

Problem PP(Ω) belongs to the general class of constrained global optimization problems, for which Horst (1990), and Horst and Tuy (1990) prescribe a variety of promising methods. These methods include branch and bound, outer approximation, and combinations of branch and bound and outer approximation techniques. In an earlier paper, Horst (1986) also presents a prototype branch and bound algorithm to solve constrained global optimization problems, and gives various sufficient conditions for convergence.

To solve PP(Ω), we propose a branch and bound algorithm which utilizes specially constructed linear bounding problems using a Reformulation Linearization Technique (RLT). In this approach, we generate nonlinear implied constraints by taking the products of bounding terms in Ω up to a suitable order, and also possibly products of other defining constraints of the problem. The resulting problem is subsequently linearized by defining new variables, one for each nonlinear term appearing in the problem. The straightforward mechanics of RLT automatically creates outer linearizations that approximate the closure of the convex hull of the feasible region. In the case of polynomial 0–1 integer programs and polynomial 0–1 mixed integer programs which are linear in the continuous variables when the 0–1 variables are fixed in values, Sherali and Adams (1989, 1990) obtain a hierarchy of such approximations leading to the exact convex hull representation of the feasible region, by using suitable applications of such an RLT procedure. For the jointly constrained biconvex programming problem, which is a special case of PP(Ω), Al-Khayyal and Falk (1983), and Al-Khayyal (1990) constructed linear bounding problems with the motivation to approximate the convex envelopes of the biconvex functions appearing in the constraints and in the objective function. By selecting proper products of lower/upper bounding constraints, their linear programming representation can equivalently be obtained via a restricted application of RLT. Sherali and Alameddine (1990) applied an extended version of RLT to derive stronger bounds in a branch and bound algorithm to solve bilinear programming problems. They also showed that for some special cases of jointly constrained bivariate bilinear programs, RLT yields the exact convex hull representation. In this paper, we generalize their branch and bound algorithm for solving continuous polynomial programs. A similar partitioning strategy is employed, involving the splitting of the set Ω into smaller hyperrectangles at each stage of the algorithm. However, the main point of difference lies in the construction of an appropriate RLT procedure used in concert with a suitable partitioning of the hyperrectangle in order to ensure the convergence of the algorithm.

The remainder of this paper is organized as follows. In the next section, we construct the RLT based linear bounding problem. Section 3 imbeds this problem in a branch and bound algorithm, and prescribes a partitioning strategy that guarantees the convergence of the overall algorithm. A third order polynomial problem is solved to illustrate the procedure in Section 4, and the final section discusses further ongoing research and implementation issues.

2. An RLT Based Linear Bounding Problem

Given Ω, in order to construct the linear programming bounding problem LP(Ω) using RLT, we begin by generating implied constraints using *distinct* products of *bounding factors* $(x_j - l_j) \geq 0$, $(u_j - x_j) \geq 0$, $j \in N$, taken δ at a time. These constraints are of the form

$$F_\delta(J_1, J_2) \equiv \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \geq 0\,, \tag{2}$$

where $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta$. The number of distinct constraints of type (2) is given by

$$\sum_{k=0}^{\delta} \binom{n+k-1}{k} \binom{n+(\delta-k)-1}{(\delta-k)}.$$

After including the constraints (2) in the problem PP(Ω), let us substitute

$$X_J = \prod_{j \in J} x_j \quad \forall J \subseteq \bar{N}\,, \tag{3}$$

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 $X_\emptyset \equiv 1$. The number of X-variables defined here, besides $X_{\{j\}}$, $j \in N$, and $X_\emptyset$ is $\binom{n+\delta}{\delta} - (n+1)$. Note that each distinct set J produces one distinct X_J variable, and that when we write $X_{(i,j,k)}$ or $X_{(J_1 \cup J_2)}$ for example, we assume that the indices within $(\cdot)$ are sequenced in nondecreasing order.

REMARK 1. *Tighter Linear Programming Representations.* Evidently, the constraints of type (2) are implied by the set Ω prior to the linearization. However, following the linearization process, these constraints impose useful interrelationships among the product variables X_J. In a likewise manner, we can also generate additional implied constraints in the form of polynomials of degree less than or equal to δ, by taking suitable products of *constraint factors* $\phi_r(x) - \beta_r \geq 0$, $r = 1, \ldots, R_1$ and/or products of bounding factors with constraint factors wherever possible. In addition, we can multiply the equality constraints $\phi_r(x) = \beta_r$, $r = R_1 + 1, \ldots, R$, defining Z, by sets of products of variables of the type $\Pi_{j \in J_1} x_j$, $J_1 \subseteq \bar{N}$, so long as the resulting polynomial expression is of degree no more than δ. Incorporating these additional constraints in LP(Ω) after making the substitutions (3), produces a tighter linear programming representation. Although this is theoretically admissible and may be computationally advantageous, it is not necessarily required for the results presented in this paper. Nevertheless, we permit the inclusion of such constraints within LP(Ω) as a user or application driven option.

Lemma 1 below verifies that LP(Ω) is indeed a relaxation of PP(Ω), and gives an important characterization of LP(Ω). Henceforth, we will let $\nu[\cdot]$ denote the value at optimality of the corresponding problem $[\cdot]$.

LEMMA 1. $\nu[\text{LP}(\Omega)] \leq \nu[\text{PP}(\Omega)]$. *Moreover, if the optimal solution* (x^*, X^*) *obtained for* LP(Ω) *satisfies* (3) *for all* $J \in \cup_{r=0}^{R} \cup_{t \in T_r} \{J_{rt}\}$, *then* x^* *solves Problem* PP(Ω).

Proof. For any feasible solution $\bar{x}$ to PP(Ω), there exists a feasible solution $(\bar{x}, \bar{X})$ to LP(Ω) with the same objective function value which is constructed using the definition (3). Hence, $\nu[\text{LP}(\Omega)] \leq \nu[\text{PP}(\Omega)]$. Moreover, if (3) holds for an optimal solution (x^*, X^*) to LP(Ω), for all $J \in \cup_{r=0}^{R} \cup_{t \in T_r} \{J_{rt}\}$, then x^* is feasible to PP(Ω), and $\nu[\text{LP}(\Omega)] = \Sigma_{t \in T_0} \alpha_{0t} X^*_{J_{0t}} = \Sigma_{t \in T_0} \alpha_{0t} [\Pi_{j \in J_{0t}} x^*_j]$, which equals the objective value of PP(Ω) at $x = x^*$. Hence, x^* solves PP(Ω). □

Notice that LP(Ω) does not explicitly contain any constraint that can be generated by constructing products of bounding factors taken less than δ at a time. The next

Lemma shows that such constraints are actually implied by those already existing in LP(Ω).

LEMMA 2. *Let* $[f(\cdot)]_l$ *denote the linearized version of a polynomial function* $f(\cdot)$ *after making the substitutions* (3). *Then, the constraints* $[F_{\delta'}(J_1, J_2)]_l \geq 0$, *where* $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta'$, $1 \leq \delta' < \delta$, *are all implied by the constraints* $[F_\delta(J_1, J_2)]_l \geq 0$ *generated via* (2), *for all distinct ordered pairs* $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta$.

Proof. For any δ', $1 \leq \delta' < \delta$, consider the surrogate of the constraints $[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_l \geq 0$ and $[F_{\delta'+1}(J_1, J_2 \cup \{t\})]_l \geq 0$, where $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta'$, and $t \in N$.

$$[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_l + [F_{\delta'+1}(J_1, J_2 \cup \{t\})]_l = [(x_t - l_t)F_{\delta'}(J_1, J_2)]_l + [(u_t - x_t)F_{\delta'}(J_1, J_2)]_l$$
$$= [x_t F_{\delta'}(J_1, J_2)]_l - l_t[F_{\delta'}(J_1, J_2)]_l + u_t[F_{\delta'}(J_1, J_2)]_l - [x_t F_{\delta'}(J_1, J_2)]_l = (u_t - l_t)[F_{\delta'}(J_1, J_2)]_l \geq 0 .$$

Since $(u_t - l_t) \geq 0$, it follows that $[F_{\delta'}(J_1, J_2)]_l \geq 0$ is implied by $[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_l \geq 0$ and $[F_{\delta'+1}(J_1, J_2 \cup \{t\})]_l \geq 0$. The required result follows by the principle of induction, and this completes the proof. □

The next result establishes an important interrelationship between the newly defined variables X_J, and prompts a partitioning strategy which drives the convergence argument.

LEMMA 3. *Let* $(\bar{x}, \bar{X})$ *be a feasible solution to* LP(Ω). *Suppose that* $\bar{x}_p = l_p$. *Then*

$$\bar{X}_{(J \cup p)} = l_p \bar{X}_J \ \forall J \subseteq \bar{N}, \quad 1 \leq |J| \leq \delta - 1 . \tag{4}$$

Similarly, if $\bar{x}_p = u_p$, *then*

$$\bar{X}_{(J \cup p)} = u_p \bar{X}_J \ \forall J \subseteq \bar{N}, \quad 1 \leq |J| \leq \delta - 1 . \tag{5}$$

Proof. First, consider the case $\bar{x}_p = l_p$. For $|J| = 1$, consider any $q \in N$ (possibly, $q \equiv p$). By Lemma 2, the following constraints are implied by $[(2)]_l$, where as before, $[\cdot]_l$ denotes the linearization of $[\cdot]$ under the substitution (3).

$$\begin{aligned} &[(x_p - l_p)(x_q - l_q)]_l = X_{(p,q)} - l_q x_p - l_p x_q + l_p l_q \geq 0 \\ &[(x_p - l_p)(u_q - x_q)]_l = -X_{(p,q)} + u_q x_p + l_p x_q - l_p u_q \geq 0 . \end{aligned} \tag{6}$$

Hence, we get

$$l_q(x_p - l_p) + l_p x_q \leq X_{(p,q)} \leq l_p x_q + u_q(x_p - l_p) . \tag{7}$$

By evaluating (7) at $(\bar{x}, \bar{X})$, we have $\bar{X}_{(p,q)} = l_p \bar{x}_q$.

Now, let us inductively assume that (4) is true for $|J| = 1, \ldots, (t-1)$, and consider $|J| = t$, where $2 \leq t \leq \delta - 1$. For any $q \in J$ (possibly $q \equiv p$), by Lemma 2, the following constraints are implied by $[(2)]_l$.

$$\begin{aligned}&\Big[(x_p - l_p)(x_q - l_q)\prod_{j\in J-q}(x_j - l_j)\Big]_l \geq 0\\&\Big[(x_p - l_p)(u_q - x_q)\prod_{j\in J-q}(x_j - l_j)\Big]_l \geq 0\,.\end{aligned}\tag{8}$$

Let us write $\Pi_{j\in J-q}(x_j - l_j) = \Pi_{j\in J-q} x_j + f(x)$, where $f(x)$ is a polynomial in x of degree no more than $t-2$. Then, from (8), we have

$$\begin{aligned}(X_{(J\cup p)} - l_pX_J) &\geq l_q(X_{(J+p-q)} - l_pX_{(J-q)}) + [l_px_qf(x) - x_px_qf(x)]_l\\&\quad + l_q[x_pf(x) - l_pf(x)]_l\\(X_{(J\cup p)} - l_pX_J) &\leq u_q(X_{(J+p-q)} - l_pX_{(J-q)}) + [l_px_qf(x) - x_px_qf(x)]_l\\&\quad + u_q[x_pf(x) - l_pf(x)]_l\,.\end{aligned}\tag{9}$$

Let $(\cdot)|_{(\bar{x},\bar{X})}$ denote the function $(\cdot)$ being evaluated at $(\bar{x}, \bar{X})$. By the induction hypothesis, $\bar{X}_{(J+p-q)} = l_p\bar{X}_{(J-q)}$, $[x_px_qf(x)]_l|_{(\bar{x},\bar{X})} = l_p[x_qf(x)]_l|_{(\bar{x},\bar{X})}$, and $[x_pf(x)]_l|_{(\bar{x},\bar{X})} = l_p[f(x)]_l|_{(\bar{x},\bar{X})}$. Hence, when we evaluate (9) at $(\bar{x}, \bar{X})$, the right hand sides of both the inequalities become zero, and this gives $\bar{X}_{(J\cup p)} = l_p\bar{X}_J$.

The case for $\bar{x}_p = u_p$ can be similarly proven by using

$$\begin{aligned}&\Big[(u_p - x_p)(u_q - x_q)\prod_{j\in J-q}(x_j - l_j)\Big]_l \geq 0\\&\Big[(u_p - x_p)(x_q - l_q)\prod_{j\in J-q}(x_j - l_j)\Big]_l \geq 0\end{aligned}\tag{10}$$

in place of (8), and this completes the proof. □

3. A Branch and Bound Algorithm

We are now ready to inbed LP(Ω) in a branch and bound algorithm to solve PP(Ω). The procedure involves the partitioning of the set Ω into subhyperrectangles, each of which is associated with a node of the branch and bound tree. Let $\Omega' \subseteq \Omega$ be such a partition. Then, LP(Ω′) gives a lower bound for the node subproblem PP(Ω′). In particular, if $(\bar{x}, \bar{X})$ solves LP(Ω′) and satisfies (3) for all $J \in \cup_{r=0}^{R} \cup_{t\in T_r}\{J_{rt}\}$, then by Lemma 1, $\bar{x}$ solves PP(Ω′), and being feasible to PP(Ω′), the value $\nu[\text{PP}(\Omega')] \equiv \nu[\text{LP}(\Omega')]$ provides an upper bound for the problem PP(Ω). Hence, we have a candidate for possibly updating the incumbent solution x^* and its value ν^* for PP(Ω). In any case, if $\nu[\text{LP}(\Omega')] \geq \nu^*$, we can fathom the node associated with Ω′. Hence, at any stage k of the branch and bound algorithm, we have a set of non-fathomed or *active nodes* denoted as (k, t), for t belonging to some index set T_k, each associated with a corresponding partition $\Omega^{k,t}$ of Ω. We now select an active node (k, t^*), $t^* \in \text{argmin}\{\nu[\text{LP}(\Omega^{k,t})], t \in T_k\}$, and proceed by decomposing the corresponding $\Omega^{k,t}$ into two subhyperrectangles, based on a *branching variable* x_p selected according to the following rule. (Here, $(\bar{x}, \bar{X})$ denotes the optimal solution obtained for $LP(\Omega^{k,t})$.)

BRANCHING RULE:

$$p \in \underset{j \in N}{\text{argmax}} \{\theta_j\},$$

$$\text{where } \theta_j = \underset{t=1,\dots,\delta-1}{\text{maximum}} \ \underset{\substack{J \subseteq \bar{N} \\ |J|=t}}{\text{maximum}} \{|\bar{X}_{(J\cup j)} - \bar{x}_j \bar{X}_J|\} \quad \text{for each } j \in N . \tag{11}$$

A formal statement of this procedure is given below.

Step 0: *Initialization*. Initialize the incumbent solution $x^* = \emptyset$, and let the incumbent objective value $\nu^* = \infty$. Set $k = 1$ and $T_k = \{1\}$. Denoting $\Omega^{1,1} \equiv \Omega$, solve $LP(\Omega^{1,1})$ to obtain an optimal solution $(\bar{x}, \bar{X}) \equiv (x^{1,1}, X^{1,1})$, and hence determine a branching variable x_p by using (11). If $\theta_p = 0$, then stop; by Lemma 1, $x^{1,1}$ solves the original problem $PP(\Omega)$. Otherwise, set $t^* = 1$, and proceed to Step 1.

Step 1: Partitioning Step (*stage k*, $k \geq 1$). Having the active node (k, t^*) to be partitioned, let x_p be its associated branching variable determined via (11). Since $\theta_p > 0$, by Lemma 3, $l_p^{k,t^*} < x_p^{k,t^*} < u_p^{k,t^*}$, where $(l_p^{k,t^*}, u_p^{k,t^*})$ are the bounds on x_p in the hyperrectangle Ω^{k,t^*}. Accordingly, partition the set Ω^{k,t^*} into two sub-hyperrectangles

$$\begin{aligned} &\Omega^{k,t_1} = \Omega^{k,t^*} \cap \{x : l_p^{k,t^*} \leq x_p \leq x_p^{k,t^*}\} \\ &\Omega^{k,t_2} = \Omega^{k,t^*} \cap \{x : x_p^{k,t^*} \leq x_p \leq u_p^{k,t^*}\} \end{aligned} \tag{12}$$

by picking indices $t_1, t_2 \not\in T_k$. After setting $T_k = (T_k - \{t\}) \cup \{t_1, t_2\}$, proceed to Step 2.

Step 2: Bounding Step. Solve the linear program $LP(\Omega^{k,t_1})$ to obtain an optimal solution $(\bar{x}, \bar{X}) \equiv (x^{k,t_1}, X^{k,t_1})$ of objective value $LB_{k,t_1} \equiv \nu[LP(\Omega^{k,t_1})]$. Using this optimal solution in (11), determine the corresponding branching variable x_p. If $\theta_p = 0$, then x^{k,t_1} solves the node subproblem $PP(\Omega^{k,t_1})$. In this case if $\nu^* > \nu[LP(\Omega^{k,t_1})]$, then update the incumbent solution $x^* = x^{k,t_1}$, and $\nu^* = LB_{k,t_1}$. Else, $\theta_p > 0$, and so store the branching variable index p to be possibly used later. Repeat Step 2 after replacing t_1 by t_2, and then proceed to Step 3.

Step 3: Fathoming Step. Fathom any nonimproving nodes by setting $T_{k+1} = T_k - \{t \in T_k : LB_{k,t} \geq \nu^*\}$. If $T_{k+1} = \emptyset$, then stop. Otherwise, update $\Omega^{k+1,t} \equiv \Omega^{k,t}$, $(x^{k+1,t}, X^{k+1,t}) \equiv (x^{k,t}, X^{k,t})$, and $LB_{k+1,t} \equiv LB_{k,t}$ for all $t \in T_{k+1}$. Increment k by 1, and proceed to Step 4.

Step 4: Node Selection Step. Select an active node (k, t^*), where $t^* \in \text{argmin}\{LB_{k,t}, t \in T_k\}$, associated with the least lower bound $LB_k \equiv LB_{k,t^*}$ over the active nodes at stage k. Return to Step 1.

THEOREM 1 (Convergence Result). *The above algorithm either terminates finitely with the incumbent solution being optimal to* $PP(\Omega)$, *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 sequence of the linear programming iterates generated at the nodes 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, and denote the associated nested sequence of partitions as $\{\Omega^{k,t(k)}\}$, $t(k) \in T_k$ for each $k \in K$, where the indices $(k, t(k))$ used to represent the nodes are selected so that

$$\mathrm{LB}_k = \mathrm{LB}_{k,t(k)} \equiv \nu[\mathrm{LP}(\Omega^{k,t(k)})] \quad \forall k \in K . \tag{13}$$

By taking any convergent subsequence of $\{x^{k,t(k)}\}$, if necessary, assume without loss of generality that

$$\{x^{k,t(k)}\}_K \to \bar{x} . \tag{14}$$

We must show that $\bar{x}$ solves PP(Ω). First of all, note that since $t(k)$ corresponds to the particular $t^* \in T_k$, for each $k \in K$, and since LB_{k,t^*} is the least lower bound at stage k, we have,

$$\nu[\mathrm{PP}(\Omega)] \geq \mathrm{LB}_{k,t(k)} = \phi_L(x^{k,t(k)}, X^{k,t(k)}) \quad \forall k \in K , \tag{15}$$

where $\phi_L(x, X)$ is the objective function of LP(Ω).

Next, observe that over the infinite sequence of nodes $\Omega^{k,t(k)}$, $k \in K$, there exists a variable x_p that is branched on infinitely often via the choice (11). Associated with x_p, there must be some index set $J_\infty \subseteq \bar{N}$, $1 \leq |J_\infty| \leq \delta - 1$, which occurs along with p infinitely often in determining θ_p. Let $K_1 \subseteq K$ be the subsequence over which $\max_{j \in N} \theta_j = \theta_p = |\bar{X}_{J_\infty \cup p} - \bar{x}_p \bar{X}_{J_\infty}|$ in (11). Then, by (11), for each $k \in K_1$, we have

$$\begin{aligned} &|X^{k,t(k)}_{(J_\infty \cup p)} - x_p^{k,t(k)} X^{k,t(k)}_{J_\infty}| \geq |X^{k,t(k)}_{(J \cup j)} - x_j^{k,t(k)} X_J^{k,t(k)}| \\ &\quad \forall J \subseteq \bar{N},\ |J| = 1, \ldots, \delta - 1, \quad j = 1, \ldots, n . \end{aligned} \tag{16}$$

Now, by the boundedness of all sequences generated, there exists a subsequence $K_2 \subseteq K_1$, such that

$$\{x^{k,t(k)}, X^{k,t(k)}, l^{k,t(k)}, u^{k,t(k)}\}_{K_2} \to (\bar{x}, \bar{X}, \bar{l}, \bar{u}) , \tag{17}$$

so that $\{\Omega^{k,t(k)}\}_{K_2} \to \bar{\Omega}$. Hence $(\bar{x}, \bar{X})$ is feasible to LP($\bar{\Omega}$). Moreover, by virtue of the partitioning (12), we know that for each $k \in K_2$, $x_p^{k,t(k)} \not\in (l_p^{k',t(k')}, u_p^{k',t(k')})$ for all $k' \in K_2$, $k' > k$, while $\bar{x}_p \in [\bar{l}_p, \bar{u}_p]$. Hence, we must have $\bar{x}_p = \bar{l}_p$ or $\bar{x}_p = \bar{u}_p$. By Lemma 3, we get

$$\bar{X}_{(J_\infty \cup p)} = \bar{x}_p \bar{X}_{J_\infty} . \tag{18}$$

But this means from (16) that as $k \to \infty$, $k \in K_2$, we have

$$\bar{X}_{(J \cup j)} = \bar{x}_j \bar{X}_J \quad \forall J \subseteq \bar{N}, \quad |J| = 1, \ldots, \delta - 1, \quad \text{and } j = 1, \ldots, n . \tag{19}$$

Hence, the definitions (3) hold true for $(\bar{x}, \bar{X})$. Therefore, $\bar{x}$ is feasible to PP(Ω), and moreover,

$$\phi_0(\bar{x}) = \phi_L(\bar{x}, \bar{X}) \geq \nu[\mathrm{PP}(\Omega)]\,. \tag{20}$$

Noting that (15) implies upon taking limits as $k \to \infty$, $k \in K_2$, that $\nu[\mathrm{PP}(\Omega)] \geq \phi_L(\bar{x}, \bar{X})$, we deduce that $\nu[\mathrm{PP}(\Omega)] = \phi_0(\bar{x})$, and so $\bar{x}$ is optimal to PP(Ω). This completes the proof. □

REMARK 2. *Special Cases*. Note that in the spirit of the foregoing algorithmic scheme, we can retain the flexibility of exploiting certain inherent special structures in designing admissible, convergent variants of this procedure. For example, consider a trilinear programming problem (see Zikan (1990) for an application in the context of tracking trajectories) in which $\delta = 3$, with any third order cross product terms being of the type $x_i x_j x_k$ for $1 \leq i \leq n_1$, $n_1 + 1 \leq j \leq n_2$, and $n_2 + 1 \leq k \leq n$, and similarly, any second order cross product term being of the form $x_i x_j$ for i and j lying in two different index sets $\{1, \ldots, n_1\}$, $\{n_1 + 1, \ldots, n_2\}$, and $\{n_2 + 1, \ldots, n\}$. For this problem, we would need to generate in (2) only those bound factor products of order 3 that involve one variable from each index set. Then, Lemma 1 holds as stated, and in Lemma 2, the corresponding second order bound factor product constraints generated by indices from two different sets are also implied. Consequently, Lemma 3 holds with $(J \cup p)$ having at most one index per index set. Accordingly, in (11), we only need to consider those $(J \cup j)$ which have at most one index from each index set, and the convergence of the resulting algorithm continues to hold by Theorem 1.

REMARK 3. *Alternate Branching Variable Selection Rule*. In light of Lemma 1 and the proof of Theorem 1, observe that we could have restricted in (11) the evaluation of only those $|\bar{X}_{(J \cup j)} - \bar{x}_j \bar{X}_J|$ quantities for which the product $\Pi_{i \in (J \cup j)} x_i$ appears in some term, or as a subset of some term, in the problem. Then, by the argument evolving around (16) in the proof of Theorem 1 and Lemma 1, the convergence of the algorithm would continue to hold.

4. An Illustrative Example

To illustrate the branch and bound algorithm of the previous section, we will solve the following nonconvex polynomial program of order $\delta = 3$.

$$\begin{aligned} \mathrm{PP}(\Omega)\text{: Minimize } \ & \phi_0(x) = x_1 x_2 x_3 + x_1^2 - 2x_1 x_2 - 3x_1 x_3 + 5x_2 x_3 \\ & \qquad - x_3^2 + 5x_2 + x_3 \\ \text{subject to } \ & 4x_1 + 3x_2 + x_3 \leq 20 \\ & x_1 + 2x_2 + x_3 \geq 1 \\ & 2 \leq x_1 \leq 5\,, \quad 0 \leq x_2 \leq 10\,, \quad 4 \leq x_3 \leq 8\,. \end{aligned} \tag{21}$$

At stage $k = 1$, $T_1 = \{1\}$, $\nu^* = \infty$, and $\Omega^{1,1} \equiv \Omega = \{x: 2 \leq x_1 \leq 5,\ 0 \leq x_2 \leq 10,\ 4 \leq x_3 \leq 8\}$. The corresponding linear program LP(Ω) has 56 constraints of type (2), linearized by using the substitution (3). Two of these constraints are given below as an example.

$$(i) \qquad J_1 = \{1, 2, 3\},\ J_2 = \emptyset: [(x_1 - 2)(x_2)(x_3 - 4)]_l \geq 0$$
$$\rightarrow X_{123} - 4X_{12} - 2X_{23} + 8x_2 \geq 0\,,$$

$$(ii) \qquad J_1 = \{1, 3\},\ J_2 = \{2\}: [(x_1 - 2)(x_3 - 4)(10 - x_2)]_l \geq 0$$
$$\rightarrow X_{123} - 4X_{12} - 10X_{13} - 2X_{23} + 40x_1 + 8x_2 + 20x_3 \leq 80\,.$$

Besides the newly generated constraints, LP($\Omega^{1,1}$) contains the original functional constraints of PP(Ω) linearized via (3), along with $\Omega^{1,1}$ itself, in its constraint set, and has the objective function

$$\phi_L(x, X) = X_{123} + X_{11} - 2X_{12} - 3X_{13} + 5X_{23} - X_{33} + 5x_2 + x_3\,.$$

Note that the entire set of variables (x, X) in the problem is given by

$$(x, X) \equiv (x_1, x_2, x_3, X_{11}, X_{12}, X_{13}, X_{22}, X_{23}, X_{33},$$
$$X_{111}, X_{112}, X_{113}, X_{122}, X_{123}, X_{133}, X_{222}, X_{223}, X_{233}, X_{333})\,.$$

Upon solving LP($\Omega^{1,1}$), we obtain,

$$(x^{1,1}, X^{1,1}) = (3, 0, 8, 8, 0, 24, 0, 0, 64, 20, 0, 64, 0, 0, 192, 0, 0, 0, 512)$$
$$\nu[\mathrm{LP}(\Omega^{1,1})] = -120\,.$$

Note that since the constraints of (21) are linear, $\bar{x}$ is feasible to (21), so that $\phi_0(x^{1,1}) = -119$ is an upper bound on the optimum to (21). Hence, the current incumbent solution is $x^* = (3, 0, 8)$ and $\nu^* = -119$. Using (11), we have, $\theta_1 = |X_{113}^{1,1} - x_1^{1,1}X_{13}^{1,1}| = 8$, and $\theta_2 = \theta_3 = 0$. (If we use Remark 3 given at the end of the previous section, then $\theta_1 = |X_{11}^{1,1} - x_1^{1,1}x_1^{1,1}| = 1$). With x_1 as the branching variable ($p = 1$), we partition $\Omega^{1,1}$ as,

$$\Omega^{1,2} = \{x: 2 \leq x_1 \leq 3, 0 \leq x_2 \leq 10, 4 \leq x_3 \leq 8\}$$
$$\Omega^{1,3} = \{x: 3 \leq x_1 \leq 5, 0 \leq x_2 \leq 10, 4 \leq x_3 \leq 8\}\,,$$

and set $T_1 = \{2, 3\}$ at Step 1. Then at Step 2, for node $(1, 2)$, LP($\Omega^{1,2}$) gives

$$(x^{1,2}, X^{1,2}) = (3, 0, 8, 9, 0, 24, 0, 0, 64, 27, 0, 72, 0, 0, 192, 0, 0, 0, 512)\,,$$
$$\nu[\mathrm{LP}(\Omega^{1,2})] = -119\,,$$

and for node $(1, 3)$, LP($\Omega^{1,3}$) gives the same solution as for LP($\Omega^{1,2}$). By using this common solution in (11), we get $\theta_1 = \theta_2 = \theta_3 = 0$. Hence, this solution is feasible to PP(Ω), but it does not improve the incumbent value. At the fathoming step (Step 3), we fathom the nodes $(1, 2)$ and $(1, 3)$, and since the list of active nodes is now empty, the solution $x^* = (3, 0, 8)$ solves the given problem (21).

Finally, let us illustrate the comment given in Remark 1. Suppose that in addition to the bound factor constraints generated above, we also generate the following constraints:

$$[(u_i - x_i)(u_j - x_j)(20 - 4x_1 - 3x_2 - x_3)]_l \geq 0 \quad \text{for} \quad 1 \leq i \leq j \leq 3 .$$

Note that this is a particular restricted set of additional constraints generated in the spirit of Remark 1. Then, it so happens that the augmented linear program $LP(\Omega^{1,1})$ itself yields the optimal solution to (21), with no branching required in this instance.

5. Conclusions and Further Considerations

In this paper we have presented a generic algorithm for globally optimizing polynomial programming problems based on the use of linear programming relaxations generated via a Reformation Linearization Technique. By incorporating appropriate bound factor products in this RLT scheme, and employing a suitable partitioning technique that is prompted by the discrepancy between the new variables and the products they represent, a convergent branch and bound algorithm has been developed.

As suggested in Remark 1, and evidenced by the foregoing illustrative example, there is a considerable flexibility, and therefore opportunity, in designing a favorable RLT process for this problem. Several types of implied constraints, or subsets, or surrogates thereof can be generated and added in a linearized form to $LP(\Omega)$, thereby tightening its representation at the expense of an increase in size. This poses an obvious question of compromise that needs to be resolved. In the very least, a successive quadratic substitution can be used as in Shor (1990) to convert the problem into an equivalent quadratically constrained quadratic problem, and then a pairwise bound factor product RLT can be employed to generate an admissible variant of our algorithm. The issue as to how this might compare with more elaborate variants is open to investigation. Moreover, as pointed out in Remark 3, different admissible branching variable selection schemes exist that need to be computationally evaluated. Also, as evident from Remark 2, special classes of polynomial programming problems might possess particular structures that can be exploited in designing special variants of the proposed algorithm. Our motivation here has been to present the basic machinery and methodology. Further investigation and experimentation is necessary to glean an adequate understanding of how best to implement this approach, depending on the actual type of problem being solved. Such issues will be pursued in a forthcoming paper.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. ECS-8807090, and the Air Force Office of Scientific Research under Grant No. 2304/B1. The Government has certain rights in this material.

References

Al-Khayyal, F.A. and J.E. Falk (1983), Jointly Constrained Biconvex Programming, *Math. of Oper. Res.* **8**, 273–286.

Al-Khayyal, F.A. (1990), Generalized Bilinear Programming: Part I. Models, Applications and Linear Programming Relaxation, Working Paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

Balas, E. and J.B. Mazzola (1984a), Nonlinear 0-1 Programming: I. Linearization Techniques, *Math. Progr.* **30**, 1–21.

Balas, E. and J.B. Mazzola (1984b), Nonlinear 0-1 Programming: II. Dominance Relations and Algorithms, *Math. Progr.* **30**, 22–45.

Floudas, C.A. and V. Visweswaran (1991), A Primal-Relaxed Dual Global Optimization Approach, Working Paper, Department of Chemical Engineering, Princeton University, Princeton, N.J. 08544-5263.

Hansen, P., B. Jaumard, and V. Mathon (1989), Constrained Nonlinear 0-1 Programming, Working paper, EUTCOR, Rutgers University, Rutgers, New Jersey.

Horst, R. (1986), A General Class of Branch-and-Bound Methods in Global Optimization with Some New Approaches for Concave Minimization, *JOTA* **51**, 271–291.

Horst, R. (1990), Deterministic Methods in Constrained Global Optimization: Some Recent Advances and New Fields of Application, *Naval Research Logistics Quarterly* **37**, 433–471.

Horst, R. and H. Tuy (1990), *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin, Germany.

Pardalos, P.M. and J.B. Rosen (1987), *Constrained Global Optimization: Algorithms and Applications*, Springer-Verlag, Berlin.

Sherali, H.D. and W.P. Adams (1990), A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems, *SIAM Journal on Discrete Mathematics* **3**, 411–430.

Sherali, H.D. and W.P. Adams (1989), A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems, Working Paper, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061.

Sherali, H.D. and A.R. Alameddine (1990), A New Reformulation-Linearization Technique for the Bilinear Programming and Related Problems, with Applications to Risk Management, Working Paper, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061.

Shor, N.Z. (1990), Dual Estimates In Multiextremal Problems, Working Paper, Institute of Cybernetics, Academy of Sciences of the Ukrainian SSR, Kiev 252207, USSR. (Presented at the IIASA Workshop on Global Optimization, Sopron, Hungary (1990)).

Zikan, K. (1990), Track Initialization and Multiple Object Tracking Problem, Working Paper, Department of Operations Research, Stanford University, Stanford, California 94305.