Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 1, No. 2, pp. 357–373, May 2005

POLYNOMIAL OPTIMIZATION: A ROBUST APPROACH

HOANG TUY

Dedicated to Professor Hiroshi Konno on the occasion of his 65th birthday.

Abstract: Existing methods of polynomial optimization actually compute an approximate optimal solution of a linear or convex relaxation of the original problem. Such an approximate optimal solution is not guaranteed to be close to the actual optimal solution, nor even to be feasible. To overcome these limitations, a robust solution approach is proposed for polynomial optimization which consists in seeking the best nonisolated feasible solution rather than the best feasible solution as usually required.

Key words: *polynomial global optimization, approximate optimal solution, Robust approach, essential optimal solution, monotonic optimization, successive incumbent transcending algorithm.*

Mathematics Subject Classification: *90C26, 90C30, 90C31, 90C57, 49K40, 65K05*

---

1 Introduction

The problem of minimizing or maximizing a multivariate polynomial under polynomial constraints is encountered in a multitude of applications: production planning, location, and distribution (see [5]), engineering design (see [3]), risk management, water treatment and distribution (see [12]), chemical process design, pooling and blending (see [4]), structural design, signal processing, robust stability analysis, design of chips, (see [2]), etc. In its general form, this problem can be formulated as :

$$\min\{p_0(x)|\ p_k(x) \geq 0\ (k = 1, \dots, m),\ a \leq x \leq b\}, \tag{P}$$

where $a, b \in \mathbb{R}^n_+$ and $p_0(x), p_1(x), \dots, p_m(x)$, are polynomials, i.e.,

$$p_k(x) = \sum_{\alpha \in A_k} c_{k\alpha} x^\alpha, \quad x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},\ c_\alpha \in \mathbb{R}, \tag{1}$$

with $\alpha = (\alpha_1, \dots, \alpha_n)$ being an n-vector of nonnegative integers. Since the set of polynomials on $[a, b]$ is dense in the space $C[a, b]$ of continuous functions on $[a, b]$ with the supnorm topology, any continuous optimization problem can be, in principle, approximated as closely as desired by a polynomial programming problem. Therefore, polynomial programming includes an extremely wide class of optimization problems. It is also a very difficult problem because many special cases of it such as the nonconvex quadratic optimization problem are known to be NP-hard.

Several methods for polynomial programming have been proposed in the last fifteen years. The earliest method is due to Shor and Stetsenko [11] (see also [12]) who were the first to point out the connection of polynomial optimization with Hilbert's 17th problem. Specifically, by reducing polynomial programming to quadratic optimization (at the expense of introducing additional variables), and using Lagrangian relaxation for estimating lower bounds, it was shown in [11] that the question of when these bounds are exact is connected with the representability of nonnegative polynomials as sums of squares of polynomial functions. It was also observed in [11] that bounds by Lagrange relaxation can be improved, sometimes drastically, by using additional constraints implied by already existing ones. In the RLT method (Reformulation-Linearization Technique) by Sherali-Adams [10], after adding all these implied constraints to the original problem, every monomial like x^α is replaced with y_α, thus transforming any polynomial in x into an affine function of (x, y). The problem then becomes a linear problem in (x, y) with the additional nonconvex constraints $y_\alpha = x^\alpha \ \forall\alpha$. By omitting these nonconvex constraints, a linear relaxation of the problem is obtained whose optimal value yields a valid lower bound. Sherali and Adams proved that if the implied constraints are suitably chosen, this bounding procedure can be combined with an exhaustive rectangular partition process to produce a convergent branch and bound algorithm. Another method of particular interest was developed in the last few years by Lasserre [6]. Using a recent result of Putinar [9] in real algebraic geometry on representation of positive polynomials, Lasserre shows that the global optimal value of problem (P) can be approximated with arbitrary accuracy by solving a suitable SDP (semidefinite program) in the variables x, y, with $y_\alpha = x^\alpha \ \forall\alpha$.

On the other hand, since any polynomial on a compact convex set can be shown to be representable as a d.c. function (difference of convex functions), it is natural that d.c. methods of optimization have been applied for solving polynomial programs. Floudas [4] developed a primal-dual relaxed algorithm called GOP for biconvex programming and extended it to a wider class of polynomial optimization problems. Maranas and Floudas [7] employed an exponential variable transformation to reduce a generalized geometric program to a d.c. optimization problem. Finally, by noting that any polynomial on $\mathbb{R}^n_+$ can be easily written as a d.m. function (difference of monotone functions), a method for solving fractional polynomial programming problems (including (P) as a special case) based on d.m. optimization was recently proposed by Tuy, Thach and Konno [14].

A common difficulty of all the above mentioned methods, except perhaps the last one [14], is that they are very sensitive to the highest degree of the polynomials involved. Even for problems of small size but with polynomials of high degree they require a huge number of additional variables. This is particularly inconvenient when dealing with signomial programming (i.e. problem (P) where the exponents $\alpha_1, \ldots, \alpha_n$ may be nonnegative rational) or with discrete constraints. But the most important, though less obvious, drawback of these methods is that they often provide only an approximate optimal value but not an approximate minimizer. For example, in the RLT or Lasserre's method, the problem (P) is reformulated as a linear program LP or a SDP in the variables x, y where $y = (y_\alpha),\ y_\alpha = x^\alpha$, such that an optimal solution $(\bar{x}, \bar{y})$ of this LP or SDP yields an approximate optimal value of (P). The trouble is that most often we do not have $\bar{x}^\alpha = \bar{y}_\alpha \ \forall\alpha$, so that $\bar{x}$ is infeasible to (P), possibly with a significant error. Moreover, in certain cases, as will be shown in Section 2 and Section 8 (Example 1), the approximate optimal value computed by these methods may be far away from the true optimal value.

Thus, despite their significant interest, these methods may not be satisfactory for the user who would prefer a good, robust, feasible solution, within an acceptable tolerance, to a solution with a better objective function value but not acceptable from the feasibility point of view or difficult to implement because of its instability. In fact, in nonconvex global optimization, so far robustness of solution methods has never been a matter of concern, despite the importance of this issue for practical implementation.

In an attempt to overcome these limitations, a general robust approach to nonconvex optimization was developed in [16]. Since polynomial programming is a special case of nonconvex optimization, this approach is naturally applicable to polynomial programming. In the present paper we will suggest a robust solution method for polynomial optimization based on the same general approach as in [16] but more easily implementable, as it requires less preliminary transformations and less additional variables.

After the Introduction, we show in Section 2 the inadequacy of the concept of approximate optimal solution as widely used in nonconvex global optimization. In Section 3 the polynomial programming problem is reformulated as a monotonic optimization problem in the form studied in a series of recent papers [13], [14], [15], [17]. In Section 4 the concept of essential optimality is introduced as a substitute for the concept of approximate optimal solution and a conceptual algorithm is proposed for finding an essential optimal solution. In Section 5, a procedure is developed for incumbent transcending, which is the core of the conceptual algorithm. In Section 6 this incumbent procedure is incorporated into the conceptual algorithm, to produce an implementable algorithm. Section 7 discusses problems where the feasible set is described by a mixed system of inequality and equality constraints. Finally, Section 8 presents some numerical examples to illustrate the basic ideas of our approach and to show how this approach works in practice.

Although we freely use ideas and methods from monotonic optimization, no familiarity with results in this field is assumed.

2 Inadequacy of Approximate Optimality

For a polynomial optimization problem (in particular, a quadratically constrained quadratic optimization problem), since the constraint set is generally nonconvex, finding a feasible solution is almost as difficult as solving the problem itself. Therefore, with rare exceptions, solution methods so far developed for problem (P) only compute either an *approximate optimal solution* (as in the standard approach), or an *approximate optimal value* (as in the RLT or Lasserre's approach).

In the standard approach, an approximate optimal solution of (P) is obtained by solving some relaxed problem. Typically, for a given tolerance $\varepsilon > 0$, the relaxed problem is taken to be

$$\min\{p_0(x)|\ p_k(x) + \varepsilon \geq 0,\ k = 1, \ldots, m,\ a \leq x \leq b\}. \qquad (\mathrm{P}(\varepsilon))$$

Furthermore, in finitely many steps, one can only guarantee an η-optimal solution of the problem $(\mathrm{P}(\varepsilon))$, i.e., a feasible solution $\tilde{x}$ to $(\mathrm{P}(\varepsilon))$ such that $p_0(\tilde{x}) \leq \min(\mathrm{P}(\varepsilon)) + \eta$ (such an $\tilde{x}$ is referred to as an (ε, η)-optimal solution of (P)). Although this has become a common practice in nonconvex global optimization, the pitfall is that an approximate optimal solution may quite often be infeasible and give an objective function value very far from the true optimal value.

Such a situation may occur when the problem has an isolated optimal solution, or there is an almost feasible but infeasible point where the objective function has a value substantially

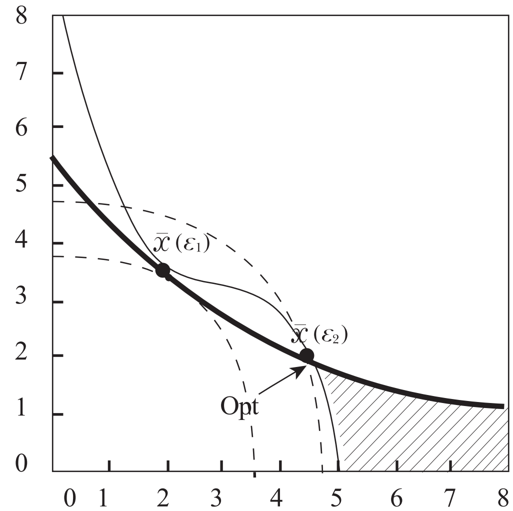

Fig. 1: Inadequate ε-approximate optimal solution

inferior to the optimal value of the problem. Consider for example the problem

$$\begin{array}{ll} \text{Minimize} & x_1^2 + x_2^2 \\ \text{s.t.} & (x_1+3)(4x_2+2) - 75 \le -10^{-5} \\ & (x_1-3)^3 + 4(x_2-3) \ge 0 \\ & 0 \le x_1 \le 8;\ 0 \le x_2 \le 8 \end{array}$$

It is easily verified that this problem is regular, i.e. its optimal value is equal to the infimum of the objective function value over the interior of the feasible set. The point $(2, 3.25)$ is infeasible but almost feasible and the value of the objective function at this point is substantially inferior to the optimal value of the problem (Fig 1). With tolerances $\varepsilon_1 = 10^{-5}, \eta = 0.01$, an (ε_1, η)-approximate optimal solution is $\bar{x}(\varepsilon_1) = (1.999988,\ 3.250007)$, close to the almost feasible solution $(2, 3.25)$, but quite far from the actual optimal solution. On the other hand, for $\varepsilon_2 = 10^{-6}, \eta = 0.01$ an (ε_2, η)-approximate optimal solution is $\bar{x}(\varepsilon_2) = (4.614169,\ 1.950098)$, very close to the true optimal solution.

Thus, even for problems without isolated feasible solutions the ε-relaxation approach may give an incorrect approximate optimal solution if ε is not sufficiently small. The trouble is that in practice we often do not know what exactly means “sufficiently small”, i.e. we do not know how small the tolerance should be to guarantee a correct approximate optimal solution.

In Lasserre’s approach, a sequence of SDP relaxations of (P) is constructed in the following way. For each fixed $l = 1, 2, \ldots$, an appropriate finite set $\mathcal{A}_l$ of n-vectors with nonnegative integral components together with a set of additional variables $y = (y_\alpha) \in \mathbb{R}^{|\mathcal{A}_l|}$ and a SDP_l in the variables (x, y) are defined such that the problem (P) is equivalent to SDP_l with the additional nonconvex constraints

$$y_\alpha = x^\alpha, \quad \alpha \in \mathcal{A}_l. \tag{2}$$

Then SDP_l is a relaxation of (P), since it is obtained from a problem equivalent to (P) by omitting the nonconvex constraints (2). Therefore, if ω_l is the optimal value of SDP_l, then

$\omega_l \leq \min(\mathrm{P})$ and the sequence $\mathcal{A}_l, l = 1, 2, \dots$, can be chosen so that

$$\omega_l \nearrow \min(\mathrm{P}) \quad (l \to +\infty). \tag{3}$$

Thus, for l satisfying $\omega_l \geq \min(\mathrm{P}) - \eta$, the value ω_l yields an approximate optimal value of (P) with tolerance η. The trouble, however, is that, for a given $\eta > 0$ we do not know how large l should be taken. Moreover, if $(\bar{x}, \bar{y})$ is an optimal solution of the relaxed problem SDP_l then, generally $\max_{\alpha \in \mathcal{A}_l} |\bar{y}_\alpha - \bar{x}^\alpha| > 0$, and so, even for η small, $\bar{x}$ may be infeasible to (P), with a notable error.

In the RLT method, for each $l = 1, 2, \dots$, a linear program LP_l in the variables (x, y) is defined such that (P) is equivalent to LP_l with the additional constraints (2). So LP_l is a linear relaxation of (P) with optimal value $\omega_l \leq \min(\mathrm{P})$ and though (3) does not generally hold, with a suitable choice of $\mathcal{A}_l$ this bound ω_l can be incorporated into a branch and bound algorithm to generate a sequence $\{(x^{(k)}, y^{(k)})\}$ such that $\max_{\alpha \in \mathcal{A}_l} |[y^{(k)}]_\alpha - [x^{(k)}]^\alpha| \to 0$ $(k \to +\infty)$ and $\bar{x} = \lim_{k \to +\infty} x^{(k)}$ is an optimal solution of (P). Again the trouble is that in most cases $x^{(k)}$ may be infeasible to (P), with a notable error even if $\max_{\alpha \in \mathcal{A}_l} |y_\alpha - x^\alpha|$ is small. A more detailed discussion on this point will be given in Section 8 for a particular implementation of the RLT approach on a specific quadratic problem.

3 Monotonic Reformulation

A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *increasing* on a box $[a, b] := \{x \in \mathbb{R}^n \,|\, a \leq x \leq b\} \subset \mathbb{R}^n$ if $f(x') \leq f(x)$ for all x', x satisfying $a \leq x' \leq x \leq b$. It is said to be a *d.m. function* on $[a, b]$ if $f(x) = f_1(x) - f_2(x)$, where f_1, f_2 are increasing on $[a, b]$.

Clearly a polynomial of n variables with positive coefficients is increasing on the orthant $\mathbb{R}^n_+$. Since every polynomial can be written as a difference of two polynomials with positive coefficients:

$$\sum_\alpha c_\alpha x^\alpha = \sum_{c_\alpha > 0} c_\alpha x^\alpha - [\sum_{c_\alpha < 0} (-c_\alpha) x^\alpha]$$

it follows that any polynomial of n variables is a d.m. function on $\mathbb{R}^n_+$.

It can be proved (see [13]) that the set $\mathrm{DM}[a, b]$ of all d.m. functions on a box $[a, b]$ forms a linear space, which is a lattice with respect to the operations

$$(g_1 \vee g_2)(x) = \max\{g_1(x), g_2(x)\}, \qquad (g_1 \wedge g_2)(x) = \min\{g_1(x), g_2(x)\}.$$

By induction it follows that if $g_1(x), \dots, g_m(x)$ are d.m. functions on $[a, b]$ then so are the functions

$$\max\{g_1(x), \dots, g_m(x)\}, \ \min\{g_1(x), \dots, g_m(x)\}.$$

As immediate consequence of this property any finite system of d.m. inequalities:

$$g_j(x) \geq 0 \quad j = 1, \dots, m$$

is equivalent to a single one $g(x) := \min_{j=1,\dots,m} g_j(x) \geq 0$. In particular, a system of polynomial constraints

$$p_j(x) \geq 0 \quad j = 1, \dots, m$$

on $\mathbb{R}^n_+$ can be replaced by a single d.m. constraint

$$g(x) := \min_{j=1,\dots,m} p_j(x) \geq 0$$

on R^n_+. If $p_0(x) = p_{01}(x) - p_{02}(x)$ where $p_{01}(x), p_{02}(x)$ are polynomials with positive coefficients, then the problem (P) can be written as

$$\begin{array}{ll} \text{minimize} & p_{01}(x) + t - p_{02}(b), \\ \text{s.t.} & p_j(x) \geq 0 \quad j = 1, \dots, m, \\ & p_{02}(x) + t - p_{02}(b) \geq 0, \\ & a \leq x \leq b,\ 0 \leq t \leq p_{02}(b) - p_{02}(a), \end{array}$$

where the function $(x, t) \mapsto p_{01}(x) + t - p_{02}(b)$ is a polynomial with positive coefficients.

It then follows that by introducing an additional variable if necessary, and changing the notation, we can convert any polynomial programming problem (P) into the form

$$\min\{f(x)|\ g(x) \geq 0,\ x \in [a, b]\}, \tag{Q}$$

where $f(x)$ is an increasing polynomial (polynomial with positive coefficients), while

$$g(x) = \min_{j=1,\dots,m}\{u_j(x) - v_j(x)\}, \tag{4}$$

and $u_j(x), v_j(x)$ are increasing polynomials.

In turn $g(x) = u(x) - v(x)$ where $u(x), v(x)$ are increasing functions (not necessarily polynomials) and the constraint $x \in [a, b], g(x) \geq 0$ is equivalent to

$$v(x) + t - u(b) \leq 0 \leq u(x) + t - u(b),\ 0 \leq t \leq u(b) - v(a).$$

Therefore, using more additional variables, a problem (Q) can be further converted into a monotonic optimization problem in the canonical form

$$\min\{F(z)|\ G(z) \leq 1 \leq H(z),\ z \in [r, s] \subset \mathbb{R}^l_+\}, \tag{CMO}$$

where F, G, H are increasing functions (which may no longer be polynomials). The robust solution method developed in [16] for polynomial programming supposes that the polynomial optimization problem has been preliminarily converted to this canonical form. Although the fact that all functions in (CMO) are increasing is an advantage, for our purpose in the sequel the form (Q) seems to be more convenient, since it requires almost no transformation and no additional variable, while preserving polynomiality of all functions involved.

4 Essential Optimal Solution

From now on we assume that the original polynomial programming problem (P) has been converted to the form (Q).

As we argued in Section 2, an algorithm giving only an ε-approximate optimal solution may not be quite correct. Furthermore, an isolated optimal solution even if computable is often difficult to implement practically because of its instability under small perturbations of the constraints. Therefore, from a practical point of view only nonisolated feasible solutions should be considered. This motivates the following definitions.

A nonisolated feasible solution x^* of (Q) is called an *essential optimal solution* if $f(x^*) \leq f(x)$ for all nonisolated feasible solutions x of (Q), i.e. if

$$f(x^*) = \min\{f(x)|\ x \in S^*\},$$

where S^* denotes the set of all nonisolated feasible solutions of (Q). Assume

$$\{x \in [a,b] |\ g(x) > 0\} \neq \emptyset. \tag{5}$$

For $\varepsilon \geq 0$, an $x \in [a,b]$ satisfying $g(x) \geq \varepsilon$ is called an *ε-essential feasible solution*, and a nonisolated feasible solution $\bar{x}$ of (Q) is called an *essential ε-optimal solution* if it satisfies

$$f(\bar{x}) - \varepsilon \leq \inf\{f(x) |\ g(x) \geq \varepsilon,\ x \in [a,b]\}. \tag{6}$$

Clearly for $\varepsilon = 0$ a nonisolated feasible solution which is essentially ε-optimal is optimal.

As a basic step towards finding an essential ε-optimal solution, let us consider the following subproblem of incumbent transcending:

($*$) *Given a real number* $\gamma \geq f(a)$ (the incumbent value)*, find a* nonisolated *feasible solution* $\hat{x}$ *of* (Q) *such that* $f(\hat{x}) \leq \gamma$*, or else establish that none such* $\hat{x}$ *exists.*

Since $f(x)$ is increasing, if $\gamma = f(b)$ then clearly an answer to the subproblem ($*$) would give a nonisolated feasible solution or else identify essential infeasibility of the problem (a problem is called *essentially infeasible* if it has no nonisolated feasible solution). If $\bar{x}$ is the best nonisolated feasible solution currently available and $\gamma = f(\bar{x}) - \varepsilon$, then an answer to ($*$) would give a new nonisolated feasible solution $\hat{x}$ with $f(\hat{x}) \leq f(\bar{x}) - \varepsilon$, or else identify $\bar{x}$ as an essential ε-optimal solution.

A finite procedure for solving the subproblem ($*$) for a given γ is referred to as a *procedure* $(*, \gamma)$. If such a procedure is available for every γ then an essential ε-optimal solution of (Q) can be found using the following algorithm.

CONCEPTUAL SIT (Successive Incumbent Transcending) ALGORITHM.

Step 0. If no nonisolated feasible solution for (Q) is known, let $\gamma = f(b)$; otherwise, let $\bar{x}$ be the best nonisolated feasible solution available, $\gamma = f(\bar{x}) - \varepsilon$.
Step 1. Call Procedure $(*, \gamma)$. If this procedure returns a nonisolated feasible solution $\hat{x}$ such that $f(\hat{x}) \leq \gamma$, go to Step 2. If it gives evidence that no nonisolated feasible solution x exists such that $f(x) \leq \gamma$, go to Step 3.
Step 2. Reset $\bar{x} \leftarrow \hat{x}$ and return to Step 0.
Step 3. Terminate: if $\gamma = f(b)$, the problem (Q) is essentially infeasible; otherwise, $\bar{x}$ is an essential ε-optimal solution.

Clearly the key to the implementation of this conceptual algorithm is the availability of a procedure $(*, \gamma)$, for any given $\gamma \geq f(a)$. In the next section we shall discuss such a procedure.

5 Procedure $(*, \gamma)$

Observe that if $g(a) > 0$ and $\gamma = f(a)$ then a solves the subproblem ($*$). Therefore, we shall assume that

$$g(a) \leq 0,\ f(a) < \gamma. \tag{7}$$

It turns out that, under this assumption, the subproblem ($*$) can be solved by solving the problem

$$\max\{g(x) |\ f(x) \leq \gamma,\ x \in [a,b]\}. \tag{Q$^*/\gamma$}$$

More precisely,

Proposition 1. *Under assumption (7):*

(i) *Any feasible solution x^0 of (Q^*/γ) such that $g(x^0) > 0$ is a nonisolated feasible solution of* (P) *with $f(x^0) \le \gamma$. In particular, if* $\max(Q^*/\gamma) > 0$ *then the optimal solution $\hat{x}$ of (Q^*/γ) is a nonisolated feasible solution of* (P) *with $f(\hat{x}) \le \gamma$.*

(ii) *If* $\max(Q^*/\gamma) < \varepsilon$ *for $\gamma = f(\bar{x}) - \varepsilon$, and $\bar{x}$ is a nonisolated feasible solution of* (P), *then it is an essential ε-optimal solution of* (P). *If* $\max(Q^*/\gamma) < \varepsilon$ *for $\gamma = f(b)$, then the problem* (P) *is ε-essentially infeasible.*

Proof. (i) In view of assumption (7), $x^0 \neq a$. Since $f(a) < \gamma$, and $x^0 \neq a$, every x in the line segment joining a to x^0 and sufficiently close to x^0 satisfies $f(x) \le f(x^0) \le \gamma,\ g(x) > 0$, i.e. is a feasible solution of (P). Therefore, x^0 is a nonisolated feasible solution of (P) satisfying $f(x^0) \le \gamma$.

(ii) If $\max(Q^*/\gamma) < \varepsilon$ then

$$\varepsilon > \sup\{g(x)|\ f(x) \le \gamma,\ x \in [a,b]\},$$

so for every $x \in [a,b]$ satisfying $g(x) \ge \varepsilon$, we must have $f(x) > \gamma = f(\bar{x}) - \varepsilon$, in other words,

$$\inf\{f(x)|\ g(x) \ge \varepsilon,\ x \in [a,b]\} > f(\bar{x}) - \varepsilon.$$

This means that, if $\bar{x}$ is a nonisolated feasible solution, then it is an essential ε-optimal solution, and if $\gamma = f(b)$, then $\{x \in [a,b]|\ g(x) \ge \varepsilon\} = \emptyset$, i.e. the problem is ε-essentially infeasible. □

BRB Algorithm for solving (Q^*/γ)

As the name (BRB, Branch-Reduce-and-Bound) indicates, this algorithm proceeds according to the standard branch and bound scheme with three basic operations: branching, reducing (the partition sets) and bounding.

-Branching consists in a successive rectangular partition of the initial box $M_0 = [a,b]$ following an *exhaustive* subdivision rule, i.e. such that any infinite nested sequence of partition sets generated through the algorithm shrinks to a singleton. A commonly used exhaustive subdivision rule is the *standard bisection*.

-Reducing consists in applying valid cuts to reduce the size of the current partition set $M = [p,q] \subset [a,b]$. The box $[p',q']$ obtained from M as a result of the cuts is referred to as a *valid reduction* of M.

-Bounding consists in estimating an upper bound $\beta(M)$ for the objective function value $g(x)$ over the feasible portion contained in the valid reduction $[p',q']$ of a given partition set $M = [p,q]$.

5.1 Valid Reduction

At a given stage of the BRB algorithm for (Q^*/γ), let $[p,q] \subset [a,b]$ be a box generated during the partitioning procedure and still of interest. The search for a nonisolated feasible solution of (Q) in $[p,q]$ such that $f(x) \le \gamma$ can then be restricted to the set $B_\gamma \cap [p,q]$, where

$$B_\gamma := \{x|\ f(x) - \gamma \le 0,\ g(x) \ge 0\}. \tag{8}$$

Since $g(x) = \min_{j=1,\dots,m}\{u_j(x) - v_j(x)\}$ with $u_j(x), v_j(x)$ being increasing polynomials (see (4)), we can also write

$$B_\gamma = \{x|f(x) \le \gamma,\ u_j(x) - v_j(x) \ge 0\ \ j = 1,\dots,m\}.$$

The reduction operation aims at replacing the box $[p, q]$ with a smaller box $[p', q'] \subset [p, q]$ without losing any point $x \in B_\gamma \cap [p, q]$, i.e. such that

$$B_\gamma \cap [p', q'] = B_\gamma \cap [p, q].$$

The box $[p', q']$ satisfying this condition is referred to as a *valid reduction* of $[p, q]$ and denoted by $\text{red}[p, q]$.

In the sequel, e^i denotes the i-th unit vector, i.e. a vector with 1 at the i-th position and 0 everywhere else.

Lemma 1. (i) *If* $f(p) > \gamma$ *or* $\min_j\{u_j(q) - v_j(p)\} < 0$, *then* $B_\gamma \cap [p, q] = \emptyset$, *i.e.* $\text{red}[p, q] = \emptyset$.
(ii) *If* $f(p) \leq \gamma$, *and* $\min_j\{u_j(q) - v_j(p)\} \geq 0$, *then* $\text{red}[p, q] = [p', q']$ *with*

$$p' = q - \sum_{i=1}^{n} \alpha_i(q_i - p_i)e^i, \quad q' = p' + \sum_{i=1}^{n} \beta_i(q_i - p'_i)e^i \tag{9}$$

where, for $i = 1, \ldots, n$,

$$\alpha_i = \sup\{\alpha|\ 0 < \alpha \leq 1,\ u_j(q - \alpha(q_i - p_i)e^i) \geq v_j(p),\ j = 1, \ldots, m\} \tag{10}$$

$$\beta_i = \sup\{\beta|\ 0 < \beta \leq 1,\ v_j(p' + \beta(q_i - p'_i)e^i) \leq u_j(q),\ j = 1, \ldots, m,\ f(p' + \beta(q_i - p'_i)e^i) \leq \gamma\}. \tag{11}$$

Proof. (i) If $f(p) > \gamma$ then $f(x) \geq f(p) > \gamma$, for every $x \in [p, q]$; if $\min_j\{u_j(q) - v_j(p)\} < 0$, then $\min_j\{u_j(x) - v_j(x)\} \leq \min_j\{u_j(q) - v_j(p)\} < 0$, for every $x \in [p, q]$. In both cases, $B_\gamma \cap [p, q] = \emptyset$.

(ii) Let $x \in [p, q]$ satisfy $u_j(x) \geq v_j(x)\ \forall j = 1, \ldots, m$. If $x \not\geq p'$ then there is i such that $x_i < p'_i = q_i - \alpha_i(q_i - p_i)$, i.e. $x_i = q_i - \alpha(q_i - p_i)$ with $\alpha > \alpha_i$. In view of (10), this implies that for some $j:\ u_j(q - (q_i - x_i)e^i) = u_j(q - \alpha(q_i - p_i)e^i) < v_j(p)$, and hence $u_j(x) \leq u_j(q - (q_i - x_i)e^i) < v_j(p) \leq v_j(x)$ (because $p \leq x \leq q - (q_i - x_i)e^i$ while $v_j(.), u_j(.)$ are increasing). Since this conflicts with $u_j(x) - v_j(x) \geq 0$, we must have $x \geq p'$, i.e. $x \in [p', q]$. Similarly, if $x \not\leq q'$ then there is i such that $x_i > q'_i = p'_i + \beta_i(q_i - p'_i)$, i.e. $x_i = p'_i + \beta(q_i - p'_i)$ with $\beta > \beta_i$ and from (11) it follows that either $f(p' + \beta(q_i - p'_i)e^i) > \gamma$, (which implies that $f(x) > \gamma$), or for some $j:\ v_j(p' + (x_i - p'_i)e^i) = v_j(p' + \beta(q_i - p'_i)e^i) > u_j(q)$, (which implies that $v_j(x) \geq v_j(p' + (x_i - p'_i)e^i) > u_j(q) \geq u_j(x)$ because $p' + (x_i - p'_i)e^i \leq x \leq q$ while $v_j(.), u(.)$ are increasing). Since in either case this conflicts with $x \in B_\gamma$, we must have $x \in [p', q']$. Therefore, $\{x \in [p, q]|\ f(x) \leq \gamma,\ u(x) - v(x) \geq 0\} \subset [p', q']$. □

Remark 1. It can easily be verified that the box $[p', q'] = \text{red}[p, q]$ still satisfies

$$f(p') \leq \gamma, \quad \min_j\{u_j(q') - v_j(p')\} \geq 0.$$

5.2 Valid Bounds

Given a box $M := [p, q]$, which is supposed to have been reduced, we want to compute an upper bound $\beta(M)$ for

$$\max\{g(x)|\ f(x) \leq \gamma,\ x \in [p, q]\}.$$

Since $g(x) = \min_{j=1,\ldots,m}\{u_j(x) - v_j(x)\}$ and $u_j(x), v_j(x)$ are increasing, an obvious upper bound is

$$\beta(M) = \min_{j=1,\ldots,m}[u_j(q) - v_j(p)]. \tag{12}$$

Though very simple, this bound suffices to ensure convergence of the algorithm, as we will see shortly. However, for a better performance of the procedure, tighter bounds can be computed using, for instance, the following

Lemma 2. (i) *If $g(p) > 0$ and $f(p) < \gamma$ then p is a nonisolated feasible solution with $f(p) < \gamma$.*

(ii) *If $f(q) > \gamma$ and $x(M) = p + \lambda(q-p)$ with $\lambda = \max\{\alpha|\ f(p+\alpha(q-p)) \le \gamma\}$, $z^i = q + (x_i(M) - q_i)e^i, \quad i = 1, \dots, n$, then an upper bound of $g(x)$ over all $x \in [p,q]$ satisfying $f(x) \le \gamma$ is*

$$\beta(M) = \max_{i=1,\dots,n} \min_{j=1,\dots,m} \{u_j(z^i) - v_j(p)\}.$$

Proof. (i) Obvious.

(ii) Let $M_i = [p, z^i] = \{x|\ p \le x \le z^i\} = \{x \in [p,q]|\ p_i \le x_i \le x_i(M)\}$. From the definition of $x(M)$ it is clear that $f(z) > f(x(M)) = \gamma$ for all $z = p + \alpha(q-p)$ with $\alpha > \lambda$. Since for every $x > x(M)$ there exists $z = p + \alpha(q-p)$ with $\alpha > \lambda$, such that $x \ge z$, it follows that $f(x) \ge f(z) > \gamma$ for all $x > x(M)$. So $\{x \in [p,q]|\ f(x) \le \gamma\} \subset [p,q] \setminus \{x|\ x > x(M)\}$. On the other hand, noting that $\{x|\ x > x(M)\} = \cap_{i=1}^n \{x|\ x_i > x_i(M)\}$, we can write $[p,q] \setminus \cap_{i=1}^n \{x|\ x_i > x_i(M)\} = \cup_{i=1,\dots,n} \{x \in [p,q]|\ p_i \le x_i \le x_i(M)\} = \cup_{i=1,\dots,n} M_i$. Since $\beta(M_i) = \min_{j=1,\dots,m}[u_j(z^i) - v_j(p)] \ge \max\{g(x)|\ x \in M_i\}$ it follows that $\beta(M) = \max\{\beta(M_i)|\ i = 1, \dots, n\} \ge \max\{g(x)|\ x \in \cup_{i=1,\dots,n} M_i\} \ge \max\{g(x)|\ x \in B_\gamma \cap [p,q]\}$. □

Remark 2. Each box $[p, z^i]$ can be reduced by the method presented above. If $[p'^i, q'^i] = \text{red}[p, z^i],\ i = 1, \dots, n$, then without much extra effort, we can have a more refined upper bound, namely

$$\beta(M) = \max_{i=1,\dots,n} \{ \min_{j=1,\dots,m} [u_j(q'^i) - v_j(p'^i)]\}.$$

The points z^i constructed as in (ii) determine a set $Z := \cup_{i=1}^n [p, z^i]$ containing $\{x \in [p,q]|\ f(x) \le \gamma\}$. In monotonic optimization such a set Z is called a *polyblock* with vertex set $z^i, i = 1, \dots, n$ [13]. The above procedure thus amounts to constructing a polyblock $Z \supset \{x \in [p,q]|\ f(x) \le \gamma\}$ – which is possible because $f(x)$ is increasing. To have a tighter bound, one can even construct a sequence of polyblocks $Z_1 \supset Z_2, \dots$, approximating the set $B_\gamma \cap [p,q]$ more and more closely. For the details of this construction the interested reader is referred to [13]. By using polyblock approximation one could compute a bound as tight as we wish; since, however, the computation cost increases rapidly with the quality of the bound, a trade-off must be made, so practically just one approximating polyblock as in the above Lemma is used.

Remark 3. In many cases, efficient bounds can also be computed by exploiting, additional structure such as partial convexity, aside from monotonicity properties (see [15]).

For example, assuming, without loss of generality, that $p \in \mathbb{R}^n_{++}$, i.e. $p_i > 0,\ i = 1, \dots, n$, and using the transformation $t_i = \log x_i, i = 1, \dots, n$, one can write each monomial x^α as $\exp\langle t, \alpha\rangle$, and each polynomial $p_k(x) = \sum_{\alpha \in A_k} c_{k\alpha} x^\alpha$ as a function $\varphi_k(t) = \sum_{\alpha \in A_k} c_{k\alpha} \exp\langle t, \alpha\rangle$. Since, as can easily be seen, $\exp\langle t, \alpha\rangle$ with $\alpha \in \mathbb{R}^m_+$ is an increasing convex function of t, each polynomial in x becomes a d.c. function of t and the subproblem over $[p,q]$ becomes a d.c. optimization problem under d.c. constraints. Here each d.c. function is a difference of two increasing convex functions of t on the box $[\exp p, \exp q]$, where $\exp p$ denotes the vector $(\exp p_1, \dots, \exp p_n)$. A bound over $[p,q]$ can then be computed by exploiting simultaneously the d.c. and d.m. structures.

6 A Robust Algorithm

Incorporating the above BRB procedure for $(\mathrm{Q}^*/\bar{x})$ into the SIT scheme yields the following robust algorithm for solving (Q):

SIT ALGORITHM FOR (Q)

Step 0. If no feasible solution is known, let $\gamma = f(b)$; otherwise, let $\bar{x}$ be the best nonisolated feasible solution available, $\gamma = f(\bar{x}) - \varepsilon$. Let $\mathcal{P}_1 = \{M_1\}, M_1 = [a, b], \mathcal{R}_1 = \emptyset$. Set $k = 1$.

Step 1. For each box $M \in \mathcal{P}_k$:

- Compute its valid reduction red$M$;
- Delete M if red$M = \emptyset$;
- Replace M by red$M$ if red$M \neq \emptyset$;
- If red$M = [p', q']$ then compute an upper bound $\beta(M)$ for $g(x)$ over the feasible solutions in red$M$. ($\beta(M)$ must satisfy $\beta(M) \le \min_{j=1,\dots,m}[u_j(q') - v_j(p')]$, see (12)). Delete M if $\beta(M) < 0$.

Step 2. Let $\mathcal{P}'_k$ be the collection of boxes that results from $\mathcal{P}_k$ after completion of Step 1. Let $\mathcal{R}'_k = \mathcal{R}_k \cup \mathcal{P}'_k$.

Step 3. If $\mathcal{R}'_k = \emptyset$ then terminate: $\bar{x}$ is an essential ε-optimal solution of (Q) if $\gamma = f(\bar{x}) - \varepsilon$, or the problem (Q) is essentially infeasible if $\gamma = f(b)$.

Step 4. If $\mathcal{R}'_k \neq \emptyset$, let $[p^k, q^k] := M_k \in \text{argmax}\{\beta(M)|\ M \in \mathcal{R}'_k\},\ \beta_k = \beta(M_k)$.

Step 5. If $\beta_k < \varepsilon$ then terminate: $\bar{x}$ is an essential ε-optimal solution of (Q) if $\gamma = f(\bar{x}) - \varepsilon$, or the problem (Q) is ε-essentially infeasible if $\gamma = f(b)$.

Step 6. If $\beta_k \ge \varepsilon$, compute $\lambda_k = \max\{\alpha|\ f(p^k + \alpha(q^k - p^k)) - \gamma \le 0\}$ and let

$$x^k = p^k + \lambda_k(q^k - p^k).$$

6. a) If $g(x^k) > 0$ then x^k is a new nonisolated feasible solution of (Q) with $f(x^k) \le \gamma$: if $g(p^k) < 0$, compute the point $\bar{x}^k$ where the line segment joining p^k to x^k meets the surface $g(x) = 0$, and reset $\bar{x} \leftarrow \bar{x}^k$; otherwise, reset $\bar{x} \leftarrow p^k$. Go to Step 7.

6. b) If $g(x^k) \le 0$, go to Step 7, with $\bar{x}$ unchanged.

Step 7. Divide M_k into two subboxes by the standard bisection (or any bisection consistent with the bounding $M \mapsto \beta(M)$). Let $\mathcal{P}_{k+1}$ be the collection of these two subboxes of M_k, $\mathcal{R}_{k+1} = \mathcal{R}'_k \setminus \{M_k\}$. Increment k, and return to Step 1.

Proposition 2. *The above algorithm terminates after finitely many steps, yielding either an essential ε-optimal solution of (Q), or an evidence that the problem is essentially infeasible.*

Proof. Since any feasible solution x with $f(x) \le \gamma = f(\bar{x}) - \varepsilon$ must lie in some box $M \in \mathcal{R}'_k$ the event $\mathcal{R}'_k = \emptyset$ implies that no such solution exists, hence the conclusion in Step 3. If Step 5 occurs, so that $\beta_k < \varepsilon$, then $\max(\mathrm{Q}^*/\gamma) \le \varepsilon$, hence the conclusion in Step 5 (see Proposition 1). Thus the conclusions in Step 3 and Step 5 are correct. It remains to show that either Step 3 ($\mathcal{R}'_k = \emptyset$) or Step 5 ($\beta_k < \varepsilon$) must occur for sufficiently large k. To this end, observe that in Step 6, since $f(p^k) \le \gamma$ (Remark 1), the point x^k exists and satisfies $f(x^k) \le \gamma$, so if $g(x^k) > 0$, then by Proposition 1, x^k is a nonisolated feasible solution with

$f(x^k) \leq f(\bar{x}) - \varepsilon$, justifying Step 6. a) Suppose now that the algorithm is infinite. Since each occurrence of Step 6a decreases the current best value at least by $\varepsilon > 0$ while $f(x)$ is bounded below it follows that Step 6. a) cannot occur infinitely often. Consequently, for all k sufficiently large, $\bar{x}$ is unchanged, and $g(x^k) \leq 0$, while $\beta_k \geq \varepsilon$. But, as $k \to +\infty$, we have, by exhaustiveness of the subdivision, $\mathrm{diam} M_k \to 0$, i.e. $\|q^k - p^k\| \to 0$. Denote by $\tilde{x}$ the common limit of q^k and p^k as $k \to +\infty$. Since

$$\varepsilon \leq \beta_k \leq \min_{j=1,\dots,m} [u_j(q^k) - v_j(p^k)],$$

it follows that

$$\varepsilon \leq \lim_{k\to+\infty} \beta_k \leq \min_{j=1,\dots,m} [u_j(\tilde{x}) - v_j(\tilde{x})] = g(\tilde{x}).$$

But by continuity, $g(\tilde{x}) = \lim_{k\to+\infty} g(x^k) \leq 0$, a contradiction. Therefore, the algorithm must be finite. □

Remark 4. The SIT Algorithm works its way to the optimum through a sequence of better and better nonisolated solutions. If for some reason the algorithm has to be stopped prematurely, some reasonably good feasible solution may have been already obtained. This is one of its advantages over most existing algorithms, which may be useless when stopped prematurely.

Remark 5. For regular problems (i.e. problems with no isolated feasible solutions), an essential optimal solution is obviously an optimal solution in the usual sense. From the computational complexity point of view the present SIT algorithm does not differ much from the monotonic branch-reduce-and-bound algorithm developed in [16].

7 Extension

I. As can easily be checked, the above method can be applied, without any modification, to solve *signomial programming* (generalized geometric programming) problems, i.e. problems of the form (P) where each vector $\alpha = (\alpha_1, \dots, \alpha_n)$ may involve rational nonintegral components (so $\alpha \in \mathbb{R}^n_+$).

II. So far we assumed (5), so that the feasible set has a nonempty interior. We now extend the method to the case when there are *equality constraints*, e.g.

$$h_l(x) = 0,\ l = 1, \dots, s\ , \tag{13}$$

(so that assumption (5) fails).

First, if some of these constraints are linear, they can be used to eliminate certain variables. Therefore, without loss of generality we can assume that all the constraints (13) are *nonlinear*. Since, however, in the most general case one cannot expect to compute a solution of a nonlinear system of equations in finitely many steps, one should be content with an approximate system

$$-\delta \leq h_l(x) \leq \delta, \quad l = 1, \dots, s\ ,$$

where $\delta > 0$ is the tolerance. In other words, a set of constraints of the form

$$\begin{array}{rcll} g_j(x) & \geq & 0 & j = 1, \dots, m \\ h_l(x) & = & 0 & l = 1, \dots, s \end{array}$$

should be replaced by the approximate system

$$\begin{aligned} g_j(x) &\geq 0 \qquad & j = 1, \dots, m \\ h_l(x) + \delta &\geq 0 & l = 1, \dots, s \\ -h_l(x) + \delta &\geq 0 & l = 1, \dots, s. \end{aligned}$$

The method presented in the previous sections can then be applied to the resulting approximate problem. With $g(x) = \min_{j=1,\dots,m} g_j(x)$, $h(x) = \max_{l=1,\dots,s} |h_l(x)|$, the required assumption is, instead of (5), $\{x \in [a,b] |\ g(x) > 0, -h(x) + \delta > 0\} \neq \emptyset$.

III. Recently, solution methods have been developed for discrete (in particular, integer) monotonic optimization [17]. It turns out that, using an appropriate modification of the concept of essential feasible solution, the present approach could be without much difficulty extended to the class of polynomial optimization problems on a discrete set. We hope to be able to discuss the details of this extension in a subsequent paper.

8 Illustrative Examples

To illustrate the basic ideas discussed above, we present some numerical examples which have been worked out with the help of N.T. Hoai Phuong. The first example is meant to show that using the concept of approximate optimal solution, one may be led to accept as optimal solution an infeasible solution with objective function value quite far from the actual optimum. The two next examples illustrate the practicability of the proposed robust method on generalized polynomial programming (signomial programming), i.e. on problems of the form (P) but with nonnegative rational exponents. Solving these problems by other existing methods, such as RLT or SDP relaxation, would require the introduction of a considerable number of additional variables and consequently, an increase in size of the problems.

Example 1. The following quadratically constrained quadratic optimization problem (Q) is studied in [1]:

$$\begin{aligned} \min_x \quad & z(x) := (12.626260(x_{12} + x_{13} + x_{14} + x_{15} + x_{16}) \\ & \qquad -1.231059(x_1x_{12} + x_2x_{13} + x_3x_{14} + x_4x_{15} + x_5x_{16}) \\ \text{s.t.} \quad & 50 \leq z(x) \leq 250; \\ & -3.475x_i + 100x_j + .0975x_i^2 - 9.75x_ix_j \leq 0 \qquad i = 1,2,3,4,5, j = i+5; \quad (*) \\ & -x_6x_{11} + x_7x_{11} - x_1x_{12} + x_6x_{12} \geq 0; \qquad (**) \\ & 50x_7 - 50x_8 - x_1x_{12} + x_2x_{13} + x_7x_{12} - x_8x_{13} = 0; \\ & 50x_8 + 50x_9 - x_2x_{13} + x_3x_{14} + x_8x_{13} - x_9x_{14} \leq 500; \\ & -50x_9 + 50x_{10} - x_3x_{14} + x_4x_{15} - x_8x_{15} + x_9x_{14} \leq 0; \\ & 50x_4 - 50x_{10} - x_4x_{15} - x_4x_{16} + x_5x_{16} + x_{10}x_{15} \leq 0; \end{aligned}$$

$50x_4 - x_4x_{16} + x_5x_{16} \geq 450;$		$-x_1 + 2x_7 \leq 1;$
$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5;$	$x_6 \leq x_7;$	$x_8 \leq x_9 \leq x_{10} \leq x_4;$
$0 \leq x_{11} - x_{12} \leq 50;$	$.001 \leq x_6 \leq 1;$	
$1 \leq x_1 \leq 8.03773157;$	$1 \leq x_7 \leq 4.51886579;$	$10^{-7} \leq x_{12} \leq 100;$
$1 \leq x_2 \leq 9;$	$1 \leq x_8 \leq 9;$	$1 \leq x_{13} \leq 50;$
$4.5 \leq x_3 \leq 9;$	$1 \leq x_9 \leq 9;$	$50 \leq x_{14} \leq 100;$
$4.5 \leq x_4 \leq 9;$	$1 \leq x_{10} \leq 9;$	$50 \leq x_{15} \leq 100;$
$9 \leq x_5 \leq 10;$	$.1 \leq x_{11} \leq 100;$	$10^{-7} \leq x_{16} \leq 50.$

The RLT method for solving this problem consists basically in the following. With the help of the additional variables $y = (y_{ij})$ where

$$y_{ij} = x_i x_j \quad (i, j = 1, \ldots, 16,\ i \leq j) \tag{14}$$

the problem is reformulated as an equivalent problem (Q') in x_i $(i = 1, \ldots, 16)$, y_{ij} $(i, j = 1, \ldots, 16,\ i \leq j)$. This problem (Q') is a linear program (LQ) with the additional nonconvex constraints (14), while the linear program (LQ) is derived from the original problem (Q) (augmented with implied constraints according to the RLT scheme) by replacing each polynomial

$$\sum_{(i,j)\in M} c_{ij} x_i x_j + \sum_{i \in N} c_i x_i^2 + \sum_i d_i x_i \tag{15}$$

with its linearization

$$\sum_{(i,j)\in M} c_{ij} y_{ij} + \sum_{i \in N} c_i y_{ii} + \sum_i d_i x_i\ . \tag{16}$$

Then a feasible solution (x, y) to (LQ) satisfying

$$|y_{ij} - x_i x_j| < \varepsilon_r \quad (i, j = 1, \ldots, 16,\ i \leq j) \tag{17}$$

is said to be an ε_r-approximate solution. In other words, if we denote by (Q'(ε_r)) the problem obtained from (Q') by replacing the constraints (14) with (17), then an ε_r-approximate solution is merely a feasible solution of (Q'(ε_r)). An ε_r-approximate solution (x^*, y^*) is said to be $(\varepsilon_r, \varepsilon_z)$-optimal if it is an ε_z-optimal solution of the problem (Q'(ε_r)). Then x^* is said to be an $(\varepsilon_r, \varepsilon_z)$-optimal solution of the original problem (Q).

In [1], by using a well devised branch and cut algorithm to solve the problem (Q'), the following $(\varepsilon_r, \varepsilon_z)$-optimal solution is found for $\varepsilon_r = \varepsilon_z = 10^{-5}$:

$$x^* = (8.03773, 8.161, 9, 9, 9, 1, 1.07026, 1.90837,\\ 1.90837, 1.90837, 50.5042, .504236, 7.26387, 50, 50, 0)$$

with objective function value $z(x^*) = 174.788$.

As argued in Section 2, since (x^*, y^*) satisfies (17) but not (14), x^* may not satisfy many constraints of (Q). In fact, it can be checked e.g. that the constraints (*) for $i = 1, 3, 4, 5$ and the constraint (**) are violated, so x^* is infeasible to the original problem (Q), and even infeasible to the approximate problem $(\tilde{Q})$ obtained from (Q) by replacing the only equality constraint $h(x) := 50x_7 - 50x_8 - x_1x_{12} + x_2x_{13} + x_7x_{12} - x_8x_{13} = 0$ with $|h(x)| \leq 0.0001$ (see Section 7.II; note that $h(x^*) = 0.000458 > 0.0001$.)

It can further be shown that $z(x^*)$ is far above the optimal value of $(\tilde{Q})$. For this we fix the variables $x_3 = x_4 = x_5 = 9,\ x_6 = 1,\ x_{14} = x_{15} = 50, x_{16} = 0$ and solve the subproblem $(\tilde{Q}^*/\gamma)$ for $\gamma = z(x^*) = 174.788$. This yields a feasible solution $\tilde{x}$ to $(\tilde{Q})$

$$\tilde{x} = (8.037732, 8.999998, 9, 9, 9, 1, 1, 1.156863,\\ 1.156863, 1.156862, 50, 0, 1, 50, 50, 0)$$

with objective function value $z(\tilde{x}) = 156.219631$ much inferior to $z(x^*) = 174.788$. Thus the $(\varepsilon_r, \varepsilon_z)$-optimal solution x^* found in [1] is infeasible to $(\tilde{Q})$ and gives an objective function value far above the optimal value of $(\tilde{Q})$.

This example illustrates the weakness of the RLT as well as Lasserre's method and also demonstrates the usefulness of the incumbent transcending approach (Procedure $(*, \gamma)$), which can help to check whether a given solution can be improved.

Example 2. (n=4)

$$\text{Minimize}\quad (3+x_1x_3)(x_1x_2x_3x_4+2x_1x_3+2)^{2/3}\quad \text{subject to}$$

$$\begin{aligned}
&-3(2x_1x_2+3x_1x_2x_4)(2x_1x_3+4x_1x_4-x_2)\\
&\quad -(x_1x_3+3x_1x_2x_4)(4x_3x_4+4x_1x_3x_4+x_1x_3-4x_1x_2x_4)^{1/3}\\
&\qquad +3(x_4+3x_1x_3x_4)(3x_1x_2x_3+3x_1x_4+2x_3x_4-3x_1x_2x_4)^{1/4}\le -309.219315\\
&-2(3x_3+3x_1x_2x_3)(x_1x_2x_3+4x_2x_4-x_3x_4)^2\\
&\quad +(3x_1x_2x_3)(3x_3+2x_1x_2x_3+3x_4)^4-(x_2x_3x_4+x_1x_3x_4)(4x_1-1)^{3/4}\\
&\qquad -3(3x_3x_4+2x_1x_3x_4)(x_1x_2x_3x_4+x_3x_4-4x_1x_2x_3-2x_1)^4\le -78243.910551\\
&-3(4x_1x_3x_4)(2x_4+2x_1x_2-x_2-x_3)^2\\
&\quad +2(x_1x_2x_4+3x_1x_3x_4)(x_1x_2+2x_2x_3+4x_2-x_2x_3x_4-x_1x_3)^4\le 9618\\
&0\le x_i\le 5\ i=1,2,3,4.
\end{aligned}$$

This problem would be difficult to solve by the RLT or Lasserre's method, since a very large number of variables would have to be introduced.

For $\varepsilon = 0.01$ the SIT Algorithm found the essential ε-optimal solution

$$x^{\text{essopt}} = (4.994594,\ 0.020149,\ 0.045424,\ 4.928073)$$

with essential ε-optimal value 5.906278 at iteration 667, and confirmed its essential ε-optimality at iteration 866.

The computation required 6.343 sec. on a PC Pentium IV 2.53GHz, RAM 256Mb DDR, and went through 53 cycles of incumbent transcending, with intermediate results for the first ten and last ten cycles as given in Table 1.

(By cycle we mean a sequence of iterations required for transcending a given incumbent; $\bar{x}$ is the new incumbent found at the end of the cycle, and Iter indicates the iteration where $\bar{x}$ is found.)

Example 3. (n=5)

$$\begin{aligned}
\text{Minimize}\quad & 4(x_1^2x_3+2x_1^2x_2x_3^2x_5+2x_1^2x_2x_3)(5x_1^2x_3x_4^2x_5+3x_2)^{3/5}\\
&\qquad +3(2x_4^2x_5^2)(4x_1^2x_4+4x_2x_5)^{5/3}
\end{aligned}$$

subject to

$$\begin{aligned}
&-2(2x_1x_5+5x_1^2x_2x_4^2x_5)(3x_1x_4x_5^2+5+4x_3x_5^2)^{1/2}\le -7684.470329\\
&2(2x_1x_2^2x_3x_4^2)(2x_1x_2x_3x_4^2+2x_2x_4^2x_5-x_1^2x_5^2)^{3/2}\le 1286590.314422\\
&0\le x_i\le 5\qquad i=1,2,3,4,5.
\end{aligned}$$

With $\varepsilon = 0.01$ the essential ε-optimal solution

$$\bar{x} = (4.987557,\ 4.984973,\ 0.143546,\ 1.172267,\ 0.958926)$$

with objective function value 28766.057367 was found at iteration 45119 and confirmed at iteration 101836. The computational time was 514.422 sec. and the maximal number of nodes of the branch and bound tree active at one iteration was 6853.

Cycle	$\bar{x}$	$f(\bar{x})$	Iter
1	(4.999998, 1.271816, 3.759639, 1.249999)	368.410194	29
2	(3.857994, 1.271816, 3.759639, 1.249999)	250.293433	30
3	(4.999998, 1.263772, 1.284919, 2.499999)	101.124566	32
4	(4.994678, 1.257941, 1.279076, 2.494081)	100.113179	33
5	(4.496902, 0.733341, 1.968319, 0.690469)	99.111902	35
6	(4.999999, 0.640076, 0.663701, 3.124999)	38.898361	53
7	(4.996012, 0.635929, 0.659562, 3.120769)	38.509352	54
8	(4.999999, 0.640076, 0.663701, 2.81250)	37.385499	56
9	(4.999999, 0.336423, 0.663701, 2.812500)	32.706640	57
10	(4.999999, 0.336423, 0.364193, 2.812500)	18.249278	58
11	(4.705440, 0.336423, 0.364193, 2.812500)	17.328990	59
. . .	. . .	. . .	. . .
44	(4.977291, 0.075587, 0.065822, 3.804033)	6.530734	307
45	(4.970758, 0.069047, 0.061690, 3.773551)	6.400772	309
46	(4.999462, 0.027206, 0.061940, 3.553025)	6.336764	375
47	(4.976755, 0.046627, 0.058138, 4.089212)	6.273397	434
48	(4.986031, 0.044224, 0.055855, 4.086719)	6.210662	435
49	(4.999631, 0.021968, 0.054891, 4.018098)	6.148556	481
50	(4.971402, 0.059281, 0.049940, 4.670158)	6.087070	531
51	(4.977333, 0.037443, 0.049073, 4.711604)	6.026199	569
52	(4.994001, 0.026386, 0.047363, 4.738176)	5.965937	609
53	(4.994594, 0.020149, 0.045424, 4.928073)	5.906278	667

Table 1

9 Conclusion

Polynomial programming forms a class of very hard nonconvex optimization problems. Most solution approaches so far developed for this class consist in computing an approximate optimal solution, which is a global η-optimal solution to some relaxation of the given problem. In this paper we demonstrated how this approach may lead to incorrect results even for regular problems, i.e. problems having a feasible set S such that $S = \mathrm{cl}(\mathrm{int})S$. To overcome the difficulty we proposed a robust solution approach which consists in seeking an essential optimal solution rather than the optimal solution. The resulting algorithm involves a sequence of successive cycles in each of which a subroutine, based on monotonic optimization, is employed for improving the current incumbent. For regular problems an η-optimal solution found by this approach is also η-optimal in the usual sense, but is always feasible, in contrast to many existing methods which may only give the η-optimal value but not necessarily a feasible solution achieving this objective function value.

Acknowledgment

The author is grateful to the referees for several useful comments and suggestions.

References

[1] C. Audet, P. Hansen, B. Jaumard, and G. Savard, A branch and cut algorithm for nonconvex quadratically constrained quadratic programming, *Math. Program.*, Ser. A, 87 (2000) 131–152.

[2] A. Ben-Tal and A. Nemirovsky, *Lectures on Modern Convex Optimization*, SIAM, MPS, Philadelphia, P.A.,2001.

[3] R. Duffin, E. Peterson and C. Zener, *Geometric Programming*, Wiley & Sons, New York, 1967.

[4] C. Floudas, *Deterministic Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1999.

[5] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed., Springer-Verlag, Berlin, 1996.

[6] J. Lasserre , Global optimization with polynomials and the problem of moments, *SIAM J. on Optimization* 11 (2001) 796–817.

[7] C. Maranas and C. Floudas, Global optimization in generalized geometric programming, *Computers and Chemical Engineering* 21 (1997) 351–370.

[8] B.A. Murtagh, and S.N. Sauders, MINOS 5.4 User' s Guide, Systems Optimization Laboratory, Dept of Operations research, Stanford University 1983.

[9] M. Putinar, Positive polynomials on compact semi-algebraic sets, *Indiana. Univ. Math. J.* 42 (1993) 969–984.

[10] H. Sherali and W. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publishers, Dordrecht, 1999.

[11] N.Z. Shor and S.I. Stetsenko, *Quadratic Extremal Problems and Nondifferentiable Optimization*, Naukova Dumka, Kiev 1989 (in Russian).

[12] N.Z. Shor, *Nondifferentiable Optimization and Polynomial Problems*, Kluwer Academic Publishers, Dordrecht, 1998.

[13] H. Tuy, Monotonic Optimization: Problems and Solution Approaches, *SIAM J. Optim.* 11: 2 (2000) 464–494.

[14] H. Tuy, P.T. Thach and H. Konno, Optimization of Polynomial Fractional Functions, *J. Global Optim.* 29 (2004) 19–44.

[15] H. Tuy, F. Al-Khayyal and P.T. Thach, Monotonic Optimization: Branch and Cut Methods, in *Essays and Surveys on Global Optimization*, C. Audet, P. Hansen and G. Savard (eds), Kluwer Academic Publishers, Dordrecht, to appear.

[16] H. Tuy, Robust Solution of Nonconvex Global Optimization problems, *Journal of Global Optimization*, to appear.

[17] H. Tuy, M. Minoux and N.T. Hoai-Phuong, Discrete Monotonic Optimization With Application to A Discrete Location Problem, *SIAM Journal on Optimization*, to appear..

Manuscript received 18 October 2004
revised 9 January 2005
accepted for publication 11 January 2005

HOANG TUY
Institute of Mathematics, 18 Hoang Quoc Viet, 10307, Hanoi, Vietnam
E-mail address: htuy@math.ac.vn