Reduction of Indefinite Quadratic Programs to Bilinear Programs

PIERRE HANSEN
GERAD, Ecole des Hautes Études Commerciales, Département des Méthodes Quantitatives et Systèmes d'Information, 5255 avenue Decelles, H3T1V6, Montréal, Canada, and RUTCOR, Rutgers University, New Jersey 08904, U.S.A.

BRIGITTE JAUMARD
Princeton University, Department of Civil Engineering and Operations Research, New Jersey 08544-5263, U.S.A., and GERAD, École Polytechnique de Montréal, Département de Mathématiques Appliquées, Succursale A, Case Postale 6079, Montréal (Québec) H3C 3A7, Canada.

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

Abstract. Indefinite quadratic programs with quadratic constraints can be reduced to bilinear programs with bilinear constraints by duplication of variables. Such reductions are studied in which: (i) the number of additional variables is minimum or (ii) the number of complicating variables, i.e., variables to be fixed in order to obtain a linear program, in the resulting bilinear program is minimum. These two problems are shown to be equivalent to a maximum bipartite subgraph and a maximum stable set problem respectively in a graph associated with the quadratic program. Non-polynomial but practically efficient algorithms for both reductions are thus obtained. Reduction of more general global optimization problems than quadratic programs to bilinear programs is also briefly discussed.

Key words. Quadratic program, bilinear program, global optimization, reduction.

1. Introduction

A *quadratic program* can be written as follows, in its most general form:

$$\text{Problem } (Q) = \begin{cases} \min \sum_{i=1}^{n} \sum_{j=i}^{n} q_{ij} x_i x_j + \sum_{i=1}^{n} q_i x_i + q_0 \\ \text{subject to:} \\ \sum_{i=1}^{n} \sum_{j=i}^{n} r_{ij}^k x_i x_j + \sum_{i=1}^{n} r_i^k x_i + r_0^k \leq 0 \quad k = 1, 2, \ldots, m \\ x_i \in \mathbb{R} \quad i = 1, 2, \ldots, n, \end{cases}$$

where the coefficients q_{ij}, q_i, q_0, r_{ij}^k, r_i^k, r_0^k $(i, j = 1, 2, \ldots, n;\ j \geq i;\ k = 1, 2, \ldots, m)$ are real numbers. No assumptions are made on convexity or concavity of the objective function or the constraints left-hand sides. The constraints possibly include nonnegativity and/or range ones. Without loss of generality, some or all of the constraints of (Q) may be assumed to be equalities. Problem (Q) thus consists in minimizing an indefinite quadratic function subject to indefinite quadratic constraints. It has numerous applications in various fields. Many of them are gathered in the recent book of Floudas and Pardalos [22].

General quadratic programs (Q) appear to be very difficult to solve exactly, i.e., as global optimization ones. Many algorithms are available for particular cases, e.g., Baron [7], Ecker and Niemi [18] and Pham [38] for convex programs; Reeves [41] for separable programs; Kough [32], Benacer and Pham Dinh [9], Pardalos, Glick and Rosen [34], Phillips and Rosen [39] for programs with indefinite objective function and linear constraints; and Al-Khayyal, Horst and Pardalos [3] for programs with concave objective function and separable constraints. Numerous algorithms have also been proposed for the minimization of concave quadratic functions subject to linear constraints, see Pardalos and Rosen [35] [36], Horst and Tuy [29] for surveys. In contrast, few papers address the general case of problem (Q), and we have found none which present a direct approach. The solution approach explored up to now consists in reducing problem (Q) to another more tractable one, and possibly deriving improved algorithms for the latter. Pham Dinh and El Bernoussi [37] and Tuy [46] express problem (Q) as a d.-c. program (i.e., a problem in which the objective function and constraints left-hand sides are differences of convex functions) and solve it by outer-approximation. It has long been known (Konno [30], Pardalos and Rosen [36]) that indefinite quadratic programs with linear constraints can be reduced to bilinear programs with separable constraints (see below) by duplication of variables. Floudas, Aggarwal and Ciric [21] extend this result, noting that problem (Q) can be reduced in a similar way to a general *bilinear program*, i.e., minimization of a bilinear function subject to bilinear constraints. Such programs may be written as follows:

$$
\text{Problem } (B) = \begin{cases} \min \sum_{i=1}^{n} \sum_{j=1}^{p} c_{ij} x_i y_j + \sum_{i=1}^{n} c'_i x_i + \sum_{i=1}^{p} c''_i y_i + c_0 \\ \text{subject to:} \\ \sum_{i=1}^{n} \sum_{j=1}^{n} a^k_{ij} x_i y_j + \sum_{i=1}^{n} a'^k_i + \sum_{i=1}^{p} a''^k_i y_i + a^k_0 \leq 0 \quad k = 1, 2, \ldots, m \\ x_i \in \mathbb{R} \quad i = 1, 2, \ldots, n, \\ y_i \in \mathbb{R} \quad i = 1, 2, \ldots, p, \end{cases}
$$

where the coefficients c_{ij}, c'_i, c''_0, a^k_{ij}, a'^k_i, a''^k_j, a^k_0 $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, p; k = 1, 2, \ldots, m)$ are real numbers. Again the constraints may include non-negativity and/or range ones, as well as equalities. Problem (B) is linear in the x variables for fixed values of the y variables, and linear in the y variables for fixed values of the x variables. It appears to have first been considered by Wolsey [48], although a particular case also involving bilinear constraints, the *modular design* problem, was previously studied by Evans [19] (see Al-Khayyal [1] for further references on that problem). The largest part of the work on bilinear programming concerns the case of separable constraints, in which the constraints are linear and decompose into two subsets involving the x or the y variables exclusively, see Pardalos and Rosen [36] for a survey. Bilinear programs with joint linear constraints are studied by Al-Khayyal and Falk [2]. Problem (B) can be solved by generalized Benders decomposition (Wolsey [48], Simões [44], Flippo [20], Floudas and Visweswaran [23], Visweswaran and Floudas [47]), branch-and-bound (Simões [44], Al-Khayyal [1]) and linearization (Sherali and Alameddine [42]).

Reduction of problem (Q) to problem (B) can be done in many ways and the choice of a way clearly influences the ease of solution of the resulting bilinear problem. In this paper we propose optimal reduction methods for two criteria: (i) the number of additional variables, discussed in Section 2, and (ii) the number of *complicating* variables (i.e., the number of variables which yield, after fixation, an easy-to-solve problem, here a linear program, see Geoffrion [26]), considered in Section 3. In the case of problem (B) the latter number is the smallest of the number of x and of y variables, omitting variables appearing only in linear terms. Reduction to problem (B) of more general global optimization problems than problem (Q) is briefly discussed in Section 4.

We only consider here reductions based on duplication of variables. While elimination of variables offers further possibilities when problem (Q) contains some linear equality constraints, we do not study them in this paper. Nor do we consider here reductions of problem (Q) to other problems than problem (B), as, e.g., biconvex programs, while acknowledging that such reductions might lead, in some cases, to easier to solve problems.

2. Short Reductions

In this section we consider reductions of Problem (Q) to problem (B) which involve the minimum number of additional variables x'_j (and new constraints $x_j = x'_j$). We call such reductions *short*. While there may be, even in the absence of linear constraints, other ways than duplication of variables to reduce problem (Q) to problem (B), which could possibly involve less additional variables *for some particular values of the coefficients*, we do not consider them here. Thus the reductions studied rely only on the *structure* of problem (Q), i.e., the information that coefficients are or are not equal to 0.

A few graph theoretic concepts will be needed, see Berge [10] for basic definitions. Let us define the *co-occurrence* graph $G = (V, E)$ of problem (Q) as follows: a vertex v_j is associated with each variable x_j $(j = 1, 2, \ldots, n)$ and an edge $\{v_i, v_j\}$ belongs to E if and only if either $q_{ij} \neq 0$ or $r^k_{ij} \neq 0$ for some k $(k = 1, 2, \ldots, m)$, i.e., if both variables x_i and x_j appear in a term of the objective function or the constraints. If this condition holds for $i = j$, i.e., $q_{ii} \neq 0$ or $r^k_{ii} \neq 0$ for some k $(k = 1, 2, \ldots, m)$, the edge is a loop. Recall that a set S of vertices of V is *stable* if no two of its vertices are adjacent, i.e., the two endpoints of an edge. The complement in V of a stable set S of G is a *transversal* of G. The *stability number* of graph G, noted $\alpha(G)$, is the maximum number of vertices in a stable set of G. A set C of vertices of V is a *clique* if any two of its vertices are adjacent. A graph G is *bipartite* it its vertex set V can be partitioned into two sets V_1 and V_2 such that any edge of E joins a vertex of V_1 to a vertex of V_2. A *subgraph* $G_A = (A, E_A)$ of G is a graph obtained by keeping all vertices of a subset A of V and all edges of E joining two vertices of A, including loops. A subgraph G_A of G is a *maximum bipartite subgraph* if it is bipartite and has a maximum number of vertices subject to that condition. Let $\alpha_2(G)$ denote this number of vertices. We call $\alpha_2(G)$ the *bipartition number* of G.

THEOREM 1. *A quadratic program* (Q) *with* n *variables and co-occurrence graph* G *with bipartition number* $\alpha_2(G)$ *has a short reduction to a bilinear program* (B) *with* $n - \alpha_2(G)$, *but no fewer, additional variables.*

Proof. Let us first show that at most $n - \alpha_2(G)$ additional variables are needed. Let $G_A = (A, E_A)$, where $A = V_1 \cup V_2$, be a maximum bipartite subgraph of the co-occurrence graph G associated with (Q). Hence, $|A| = \alpha_2(G)$. Let $W_1 = V \backslash A$. Duplicate the vertices of W_1 and denote by W_2 the corresponding vertex set, i.e., $W_2 = \{v_j' \mid v_j \in W_1\}$. Construct a new graph $G' = (V \cup W_2, E')$ where $E' = E_A \cup E_1 \cup E_2 \cup E_3$ with $E_1 = \{\{v_i, v_j\} \in E \mid v_i \in V_2$ and $v_j \in W_1\}$; $E_2 = \{\{v_i, v_j'\} \mid \{v_i, v_j\} \in E,\ v_i \in V_1$ and $v_j \in W_1\}$ and $E_3 = \{\{v_j, v_j'\} \mid v_j \in W_1$ and there is a loop at vertex $v_j\} \cup \{\{v_i, v_j'\} \mid i < j,\ \{v_i, v_j\} \in E,\ v_i \in W_1$ and $v_j \in W_1\}$. G'' is a bipartite graph with edges joining vertices of $V_1 \cup W_1$ to vertices of $V_2 \cup W_2$. Note that each product of variables (including squares) in problem (Q) is associated with an edge of G'. Whenever a product $x_i x_j$ is associated with an edge $\{v_i, v_j'\}$ of $E_2 \cup E_3$, replace x_j by x_j' and add the constraint $x_j = x_j'$. This yields a bilinear program with as variables of the first set (variables x in the definition of program (B)) those associated with vertices of $V_1 \cup W_1$, and as variables of the second set (variables y in the definition of program (B)) those associated with vertices of $V_2 \cup W_2$.

We now show that at least $n - \alpha_2(G)$ additional variables are needed. Assume problem (Q) can be reduced to problem (B) by introducing less than $n - \alpha_2(G)$ additional variables. Consider then the subgraph $G_D = (D, E_D)$ of the co-occurrence graph G associated with (Q), induced by the set D of vertices associated with variables which have not been duplicated in the reduction. As G_D is associated with a bilinear subproblem (i.e., a problem obtained by deleting some terms in the objective function and/or constraints of the resulting problem (B)), it must be bipartite. However, by the above assumption, $|D| > \alpha_2(G)$, a contradiction. ∎

Finding a maximum bipartite subgraph of a graph G is NP-hard (Garey, Johnson and Stockmeyer [25]). However, this problem can be solved in practice for graphs of moderate size. Indeed, first note that additional variables must be introduced for all squared variables. The corresponding vertices in G may therefore be deleted. Finding $\alpha_2(G)$ in the resulting graph can then be reduced to a maximum stable set problem by using Theorem 13, Chapter 16 of Berge [10]. Two copies of the graph are made and homologous vertices v_j and $\tilde{v}_j$ are joined by edges. This leads to a graph $G + K_2$, the *sum* of graph G and an edge, according to the definition of the graph sum operation (see Berge [10], p. 304). Then any maximum stable set in graph $G + K_2$ is associated with a maximum bipartite subgraph of G. Recent algorithms for the maximum clique or stable set problem (Carraghan and Pardalos [13], Friden, Hertz and de Werra [24], Balas and Yu [6]) allow the solution of problems with several hundreds of variables.

EXAMPLE 1. (Colville problem 3 [14], Hock and Schittkowski problem 83 [28]).

$$\text{Problem } (Q_1) \begin{cases} \min c_1x_3^2 + c_2x_1x_5 + c_3x_1 - c_4 \\ \text{subject to:} \\ 0 \leqslant a_1 + a_2x_2x_5 + a_3x_1x_4 - a_4x_3x_5 \leqslant 92 \\ 90 \leqslant a_5 + a_6x_2x_5 + a_7x_1x_2 + a_8x_3^2 \leqslant 110 \\ 20 \leqslant a_9 + a_{10}x_3x_5 + a_{11}x_1x_3 + a_{12}x_3x_4 \leqslant 25 \\ l_j \leqslant x_j \leqslant u_j \quad j = 1, 2, 3, 4, 5 . \end{cases}$$

Coefficients c_k $(k = 1, 2, 3, 4)$ and a_k $(k = 1, 2, \ldots, 12)$ as well as bounds l_j and u_j are positive; values are given in Hock and Schittkowski [28]. The co-occurrence graph G_1 of problem (Q_1) is represented on Figure 1. As it contains a loop at vertex v_3, variable x_3 must be duplicated. Vertex v_3 can thus be omitted when determining a maximum bipartite subgraph of G_1.

The sum graph $G_1 + K_2$ defined to determine $\alpha_2(G)$ is represented on Figure 2. Vertices of a maximum stable set, i.e., v_1, $\tilde{v}_2$ and $\tilde{v}_4$, are underlined. A maximum

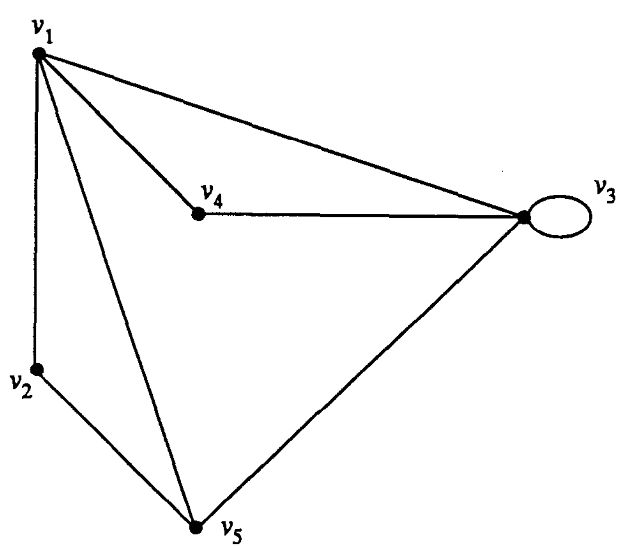

Fig. 1. Co-occurrence graph G_1 for Example 1.

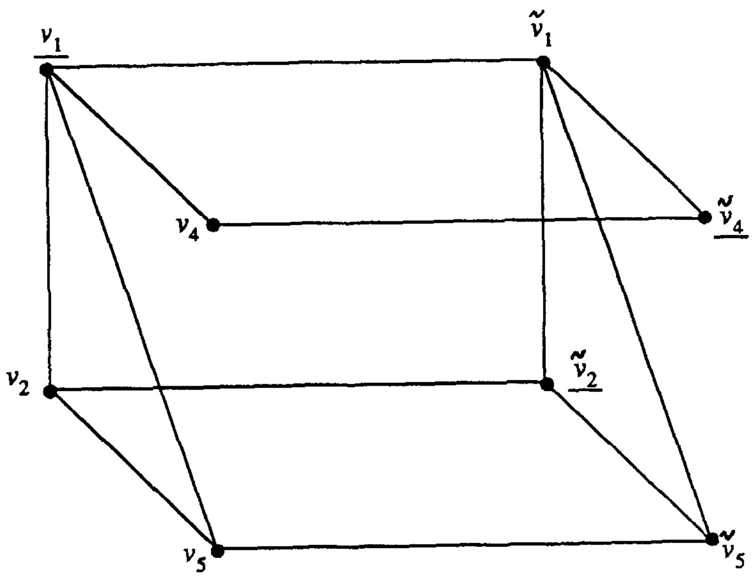

Fig. 2. Sum graph $G_1 + K_2$ for Example 1.

bipartite subgraph of G is therefore induced by vertices v_1, v_2 and v_4. This leads to $V_1 = \{v_2, v_4\}$, $V_2 = \{v_1\}$, $W_1 = \{v_3, v_5\}$, and $W_2 = \{v_3', v_5'\}$. The graph G_1' associated with the resulting bilinear problem (B_1) is depicted on Figure 3. A short reduction of problem (Q_1) is obtained by duplicating variables x_3 and x_5; variables x_1, x_3' and x_5' are complicating. After renaming some of the variables $(x_1 \leftarrow y_1;\ x_5 \leftarrow x_1;\ x_3' \leftarrow y_3;\ x_5' \leftarrow y_2)$, the resulting bilinear Problem (B_1) may be written:

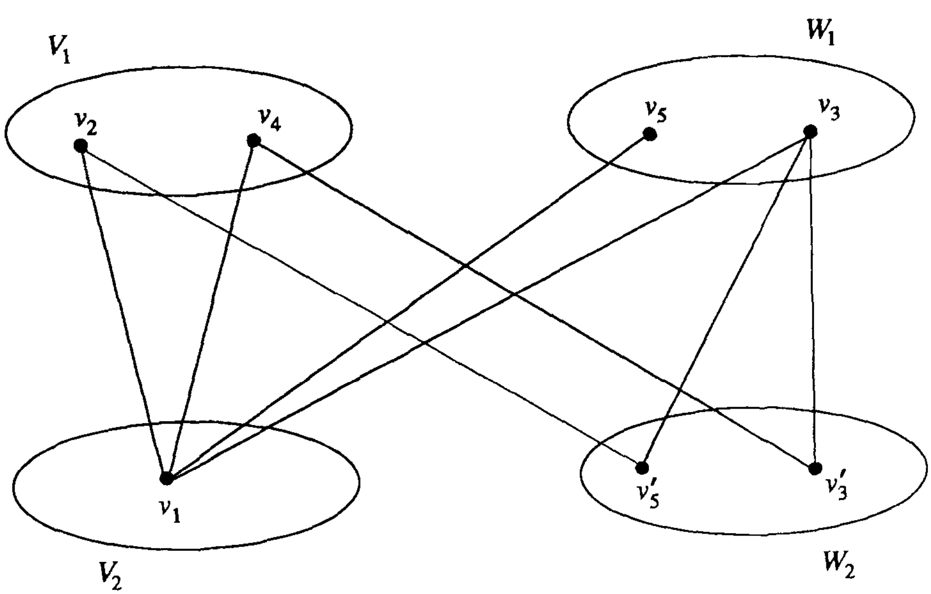

Fig. 3. Graph G_1' for Example 1.

$$\text{Problem } (B_1)\begin{cases} \min c_1x_3y_3 + c_2x_1y_1 + c_3y_1 - c_4 \\ \text{subject to:} \\ 0 \leqslant a_1 + a_2x_2y_2 + a_3x_4y_1 - a_4x_3y_2 \leqslant 92 \\ 90 \leqslant a_5 + a_6x_2y_2 + a_7x_2y_1 + a_8x_3y_3 \leqslant 110 \\ 20 \leqslant a_9 + a_{10}x_3y_2 + a_{11}x_3y_1 + a_{12}x_4y_3 \leqslant 25 \\ x_1 = y_2 \\ x_3 = y_3 \\ l_5 \leqslant x_1 \leqslant u_5 \\ l_j \leqslant x_j \leqslant u_j \quad j = 2, 3, 4 \\ l_1 \leqslant y_1 \leqslant u_1 . \end{cases}$$

3. Narrow Reduction

We now consider reductions of problem (Q) to a problem (B) with the minimum number of complicating variables. We call such reductions *narrow*. Again we consider only reductions relying solely on the structure of problem (Q).

THEOREM 2. *A quadratic program (Q) with n variables and co-occurrence graph G with stability number $\alpha(G)$ has a narrow reduction to a bilinear program (B) with $n - \alpha(G)$, but no fewer, complicating variables.*

Proof. Let us first show that at most $n - \alpha(G)$ complicating variables are needed. Let $S \subseteq V$ denote a maximum stable set of G. Hence, $|S| = \alpha(G)$. Let $W_1 = V \backslash S$. Duplicate the vertices of W_1 and denote by W_2 the corresponding vertex set, i.e., $W_2 = \{v_j' \mid v_j \in W_1\}$. Construct a new graph $G' = (V \cup W_2, E')$ where $E' = E_1 \cup E_2 \cup E_3$ with $E_1 = \{\{v_i, v_j\} \in E \mid v_i \in S \text{ and } v_j \in W_1\}$; $E_2 = \{\{v_i, v_j'\} \mid \{v_i, v_j\} \in E,\ i > j,\ v_i \in W_1 \text{ and } v_j \in W_1\}$ and $E_3 = \{\{v_j, v_j'\} \mid v_j \in W_1$ and there is a loop at vertex v_j in $G\}$.

Each product of variables (including squares) in problem (Q) is associated with an edge of G'. Whenever a product x_ix_j is associated with an edge $\{v_i, v_j'\}$ of $E_2 \cup E_3$, replace x_j by x_j' and add the constraint $x_j = x_j'$. This yields a bilinear program with as variables of the first set (variables x in the definition of program (B)) those associated with the vertices of W_1 and as variables of the second set (variables y in the definition of program (B)) those associated with the vertices of $W_2 \cup S$. Thus, at most $|W_1| = n - \alpha(G)$ complicating variables are needed.

We now show that at least $n - \alpha(G)$ complicating variables are required. Consider a narrow reduction of problem (Q) and an associated bipartite graph $G' = (V_1 \cup V_2, E')$ where V_1 is the set of vertices associated with the variables of the first set in program (B) and V_2 is the set of vertices associated with the variables of the second set. Hence $|V_1| \leqslant |V_2|$. Duplicating a vertex v_j of a graph G by a vertex v_j', while keeping the same set of edges (those incident with v_j in the original graph being incident either with v_j or with v_j') increases $\alpha(G)$ by at most 1. By iteration, if q is the number of additional variables needed, $\alpha(G') \leqslant \alpha(G) + q$. As V_2 is stable, it follows that $|V_1| \geqslant n + q - \alpha(G')$ since $|V_1 \cup V_2| = n + q$. Hence $|V_1| \geqslant n + q - (\alpha(G) + q) = n - \alpha(G)$. ∎

EXAMPLE 2. (Dembo [15] [16]).

$$\text{Problem } (Q_2) \begin{cases} \min c_1x_{11} + c_2x_{12} + c_3x_{13} \\ \text{subject to:} \\ c_4x_8 + c_5x_1x_8 - x_{11} \leq 0 \\ c_6x_9 + c_7x_2x_9 - x_{12} \leq 0 \\ c_8x_{10} + c_9x_3x_{10} - x_{13} \leq 0 \\ c_{10}x_2 + c_{11}x_2x_5 + c_{12}x_2^2 - x_5 \leq 0 \\ c_{13}x_3 + c_{14}x_3x_6 + c_{15}x_3^2 - x_6 \leq 0 \\ c_{16}x_1x_8 + c_{17}x_4x_7 + c_{18}x_4x_8 - x_5x_7 \leq 0 \\ c_{19}x_2x_9 + c_{20}x_5x_8 + c_{21}x_6 + c_{22}x_5 + c_{23}x_1x_8 + c_{24}x_6x_9 \leq 1 \\ c_{25}x_3x_{10} + c_{26}x_6x_9 + c_{27}x_2 + c_{28}x_2x_{10} + c_{29}x_6 - x_2x_9 \leq 0 \\ c_{30} + c_{31}x_2x_{10} + c_{32}x_3x_{10} - x_2 \leq 0 \\ c_{33}x_2 - x_3 \leq 0 \\ c_{34}x_1 - x_2 \leq 0 \\ c_{35}x_7 + c_{36}x_8 \leq 1 \\ c_{37}x_1 + c_{38}x_1x_4 + c_{39}x_1^2 - x_4 \leq 0 \\ l_j \leq x_j \leq u_j \quad j = 1, 2, \ldots, 13 . \end{cases}$$

Coefficients c_k for $k = 1, 2, \ldots, 39$ include positive and negative values and all bounds l_j and u_j are strictly positive, see Dembo [16].

The co-occurrence graph G_2 of Problem (Q_2) is represented in Figure 4. Vertices belonging to a maximum stable set, i.e., v_4, v_5, v_6, v_{10}, v_{11}, v_{12}, and v_{13}, are underlined. The stability number $\alpha(G)$ is equal to 7. The bipartite graph G_2' deduced from G_2 and its stable set are represented on Figure 5. As v_7', v_8' and v_9' are isolated vertices, only $|W_2| - 3 = 3$ additional variables are required. So program (Q_2) can be reduced to a bilinear program (B_2) with 6 complicating

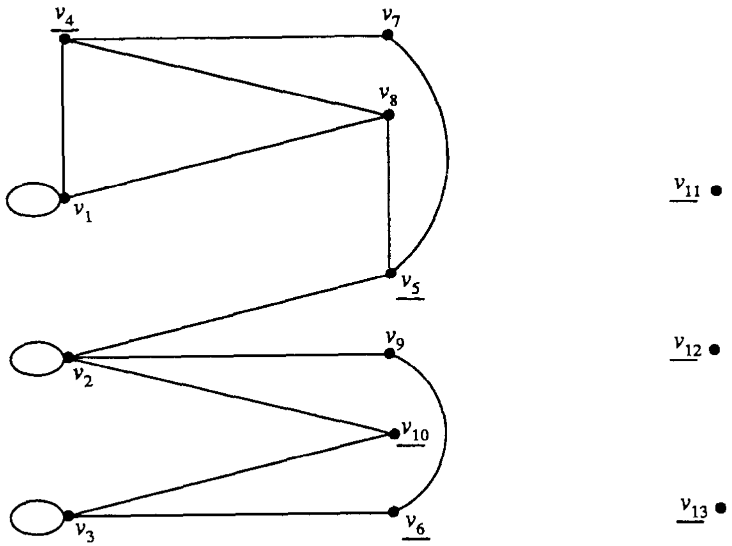

Fig. 4. Co-occurrence graph G_2 for Example 2.

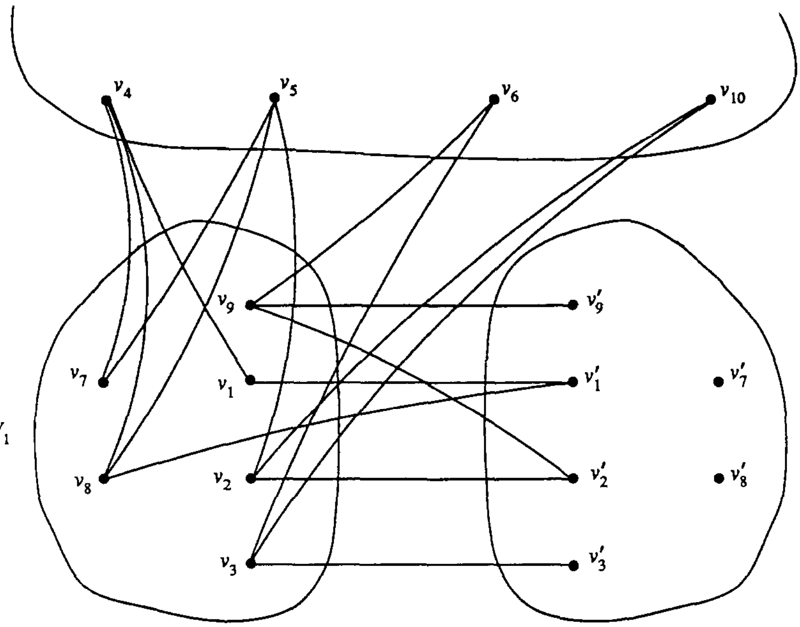

Fig. 5. Graph G'_1 for Example 1.

variables (those associated with vertices of W_1) and 3 additional variables (those associated with vertices of W_2). After renaming some of the variables ($x_4 \leftarrow y_4$; $x_5 \leftarrow y_5$; $x_6 \leftarrow y_6$; $x_7 \leftarrow x_4$; $x_8 \leftarrow x_5$; $x_9 \leftarrow x_6$; $x_{10} \leftarrow y_7$; $x_{11} \leftarrow y_8$; $x_{12} \leftarrow y_9$; $x_{13} \leftarrow y_{10}$), program (B_2) may be written as follows:

$$
\text{Problem } (B_2)\ \begin{cases}
\min c_1 y_8 + c_2 y_9 + c_3 y_{10} \\
\text{subject to:} \\
c_4 x_5 + c_5 y_1 x_5 - y_8 \leqslant 0 \\
c_6 x_6 + c_7 y_2 x_6 - y_9 \leqslant 0 \\
c_8 y_7 + c_9 y_7 x_3 - y_{10} \leqslant 0 \\
c_{10} x_2 + c_{11} y_5 x_2 + c_{12} x_2 y_2 - y_5 \leqslant 0 \\
c_{13} x_3 + c_{14} y_6 x_3 + c_{15} x_3 y_3 - y_6 \leqslant 0 \\
c_{16} y_1 x_5 + c_{17} y_4 x_4 + c_{18} x_5 y_4 - y_5 x_4 \leqslant 0 \\
c_{19} y_2 x_6 + c_{20} y_5 x_5 + c_{21} y_6 + c_{22} y_5 + c_{23} y_1 x_5 + c_{24} y_6 x_6 \leqslant 0 \\
c_{25} y_7 x_3 + c_{26} y_6 x_6 + c_{27} x_2 + c_{28} x_2 y_7 + c_{29} y_6 - y_2 x_6 \leqslant 0 \\
c_{30} + c_{31} x_2 y_7 + c_{32} x_3 y_7 - x_2 \leqslant 0 \\
c_{33} x_2 - x_3 \leqslant 0 \\
c_{34} x_1 - x_2 \leqslant 0 \\
c_{35} x_4 + c_{36} x_5 \leqslant 1 \\
c_{37} x_1 + c_{38} x_1 y_4 + c_{39} x_1 y_1 - y_4 \leqslant 0 \\
x_1 = y_1 \\
x_2 = y_2 \\
x_3 = y_3 \\
\text{and the corresponding range constraints.}
\end{cases}
$$

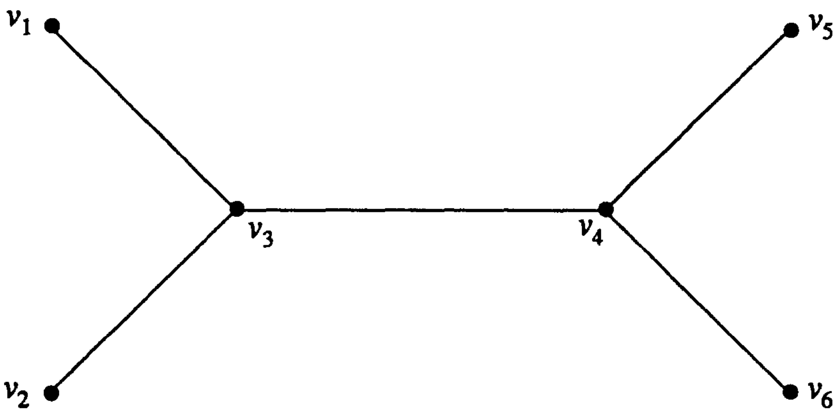

Fig. 6. Co-occurrence graph G for Example 3.

The narrow reduction so obtained is also short, as at least three additional variables are needed since three variables are squared in program (Q_2). Note, however, that, in some cases, there may be no narrow reduction among the class of short reductions. Moreover, a program which is already in bilinear form may still have a non-trivial narrow reduction, which will be obtained by applying Theorem 2. In other words, adding variables to a bilinear program may reduce the number of complicating variables. This is illustrated by the following example:

EXAMPLE 3. Problem (Q_3). $\min x_1y_1 + x_2y_1 + x_3y_1 + x_3y_2 + x_3y_3$.

This program has 3 complicating variables, e.g., x_1, x_2 and x_3. Its co-occurrence graph G_3 is represented on Figure 6 with vertices $v_1, v_2, v_3, v_4, v_5, v_6$ associated respectively with variables x_1, x_2, y_1, x_3, y_2 and y_3. Graph G_3 has a minimum transversal (the complement of a maximum stable set) with 2 vertices, i.e. v_3 and v_4, associated with variables y_1 and x_3. Duplicating x_3 yields a problem with 7 variables but only 2 complicating variables. After renaming the variables $(y_1 \leftarrow x_1;\ y_2 \leftarrow y_4;\ y_3 \leftarrow y_5;\ x_1 \leftarrow y_1;\ x_2 \leftarrow y_2;\ x_3 \leftarrow x_2;\ x_3' \leftarrow y_3)$, it may be written:

$$\text{Problem } (B_3)\begin{cases} \min x_1y_1 + x_1y_2 + x_1y_3 + x_2y_4 + x_2y_5 \\ \text{subject to:} \\ x_2 = y_3 . \end{cases}$$

4. Extensions

Many more general problems than quadratic programs (Q) can be reduced to bilinear programs (B). In this Section, we briefly discuss how this can be done.

4.1. POLYNOMIAL PROGRAMS

As noted by Floudas, Aggarwal and Ciric [21], polynomial programs can be reduced to bilinear programs (B) by repeated substitutions of squares of variables or products of two variables by additional variables. Note that substitutions of additional variables to products of more than 2 variables or to variables raised to powers higher than 2 could also be used. They sometimes lead to shorter reductions as shown in the Example 4 below. We first consider the case of a single product of variables:

PROPOSITION 1. *Minimization of a product of k ($k \geq 3$) variables can be reduced to a bilinear program (B) with $k-1$ complicating variables by introducing $k-2$ additional variables. Moreover, both of these numbers are minimum.*

Proof. We first show that at most $k-1$ complicating variables and $k-2$ additional variables are needed to obtain a bilinear program. Let $\Pi_{i=1}^{k} x_i$ be the product to be minimized: define $k-2$ new variables by $y_1 = x_1x_2$, $y_j = y_{j-1}x_{j+1}$ for $j = 2, 3, \dots, k-2$; then the product may be written as $y_{k-2}x_k$. Moreover, the set $\{x_1, y_1, y_2, \dots, y_{k-2}\}$ can be chosen as set of complicating variables.

We now show that at least $k-1$ complicating variables and $k-2$ additional variables are required. At each successive substitution reduces the number of variables in the product by one, $k-2$ additional variables are needed. Moreover, it is easy to show that each initial or additional variable must appear in exactly one product of the resulting bilinear program (B). So the co-occurrence graph G of program (B) is a perfect matching on $2k-2$ vertices and has a minimum transversal containing $n - \alpha(G) = k-1$ vertices. ■

An immediate application of Proposition 1 is to *multiplicative programs* (Konno and Kuno [31], Thoai [45]) in which the objective function is a product of $k \geq 2$ linear functions and the constraints are linear or convex. If $k = 2$, the most frequently studied case, two variables are set equal to the linear factors and any one of them can be chosen as unique complicating variable when constraints are linear.

PROPOSITION 2. *Minimization of a product $\Pi_{j\in J} x_j^{p_j}$ of variables raised to positive integer powers p_j (with at least $p_j \geq 2$ for some $j \in J$) can be reduced to a bilinear program with $\Sigma_{j\in J} p_j - 2 + \max\{2 - |J|, 0\}$ additional variables or with $|J|$, but no fewer than $|J| - 1$, complicating variables.*

Proof. Let us first write $\Pi_{j\in J} x_j^{p_j}$ as a product of $\Sigma_{j\in J} p_j$ variables, with repetitions. Then if $|J| > 1$ a similar reasoning to the proof of Proposition 1 shows that $\Sigma_{j\in J} p_j - 2$ additional variables suffice; if $|J| = 1$ (i.e., $J = \{j\}$) $p_j - 1$ additional variables suffice. Again from Proposition 1 at least $|J| - 1$ complicating variables are needed. Reducing separately each $x_j^{p_j}$ with $p_j \geq 2$ in the product yields a bilinear program with $|J|$ complicating variables, i.e., the variables x_j themselves. ■

The bound on the number of additional variables of Proposition 2 is not always sharp, as we now show:

EXAMPLE 4. $\min x_1^2 x_2^2 x_3^2$.

The reduction of Proposition 1 leads to, e.g., $y_1 = x_1 x_2$, $y_2 = x_1 y_1$, $y_3 = x_2 y_2$, $y_4 = x_3 y_3$ and the product becomes equal to $x_3 y_4$, with 4 additional variables. However, the reduction $y_1 = x_1 x_2$, $y_2 = x_3 y_1$, $y_3 = x_3 y_1$ makes the product equal to $y_2 y_3$ and uses only 3 additional variables.

Let us now consider a general *polynomial program*:

$$\text{Problem } (P) \begin{cases} \min \sum_{k=1}^{K_0} c_k \prod_{j \in J_k} x_j^{p_j} \\ \text{subject to:} \\ \sum_{k=1}^{K_i} a_{ik} \prod_{j \in J_{k_i}} x_j^{p_{ij}} \leqslant 0 \quad i = 1, 2, \ldots, m, \end{cases}$$

where the c_k for $k = 1, 2, \ldots, K_0$ and a_{ik} for $k = 1, 2, \ldots, K_i$ and $i = 1, 2, \ldots, m$ are real numbers, the p_j and p_{ij} for the same index sets are positive integers. Again the constraints may contain non-negativity or range ones as well as equalities.

In order to partially extend Propositions 1 and 2 to the case of program (P), we need some more definitions. Recall that a *hypergraph* $H = (V, E)$ (cf Berge [11]) is a finite set of subsets E_i (called *edges*) of a set V of elements (called *vertices*). The *co-occurrence hypergraph* H of program (P) is defined by associating vertices $v_j \in V$ with variables x_j ($j = 1, 2, \ldots, n$) and taking as edges the vertex sets corresponding to the variables of each product of variables in (P) (without repetition). The *2-section of a hypergraph* H is the graph $H_2 = (V, E(H))$ obtained by taking the same set V of vertices as in H and as edges all pairs of vertices belonging both to the same edge of H. A partial subgraph $C = (V, E_C)$ of H_2 will be called an *edge-edge covering* of H if the vertex set of each edge E_i of H contains the endpoints $|E_i| - 2$ edges of C which do not induce any cycle (in other words, each edge must contain one or two disjoint trees of C with a total length of $|E_i| - 2$). Note than any reduction of problem (P) to a bilinear program (B) induces an edge-edge covering of H. Edges correspond to pairs of vertices associated with variables x_k, x_l for substitutions of the form $y_j = y_k x_l$, and recursively to the highest indexed variable in the expression of y_k or y_l for substitutions of the form $y_j = x_k x_l$ or $y_j = y_k y_l$.

PROPOSITION 3. *Let (P) be a polynomial program with n variables of which t have degree greater than or equal to 2, and a co-occurrence hypergraph H. Let $\mathscr{C}$ denote the set of edge-edge coverings of H. Then the number of additional variables in any reduction of (P) to a bilinear program (B) is at least*

$$t + \min_{C \in \mathscr{C}} |E_C| .$$

Proof. Terms of the form $x_j^{p_j}$ or $x_j^{p_{ij}}$ with $p_j \geq 2$ or $p_{ij} \geq 2$ may be reduced first. This takes at least t additional variables and does not modify the co-occurrence hypergraph H. Then, to reduce terms with more than 2 variables, additional variables corresponding to an edge-edge covering of H must be used. There are at least $\min_{C \in \mathscr{C}} |E_C|$ of them. ∎

The bound of Proposition 3 is often loose, but is sufficient to show that some reductions of polynomial to bilinear programs are short.

EXAMPLE 5. (Bartholomew-Biggs [8], Hock and Schittkowski [28] problem 71).

$$\text{Problem } (P_1) \begin{cases} \min x_1^2 x_4 + x_1 x_2 x_4 + x_1 x_3 x_4 + x_3 \\ \text{subject to:} \\ x_1 x_2 x_3 x_4 - 25 \geq 0 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 - 40 = 0 \\ 1 \leq x_j \leq 5 \quad j = 1, 2, 3, 4 . \end{cases}$$

The co-occurrence hypergraph of problem (P_1) is represented on Figure 7; its 2-section is the complete graph on four vertices. Four additional variables, x_1', x_2', x_3', and x_4', are needed to reduce the squared variables. A minimum edge-edge covering of the 2-section H_2 of H is represented in dash lines on Figure 7, it consists of edges $\{v_1, v_2\}$ and $\{v_1, v_4\}$.

Thus, two additional variables $y_1 = x_1 x_4$ and $y_2 = y_1 x_2$ suffice to reduce all products to bilinear ones. The co-occurrence graph so obtained is bipartite. The reduction uses 6 additional variables and is short. The resulting bilinear program

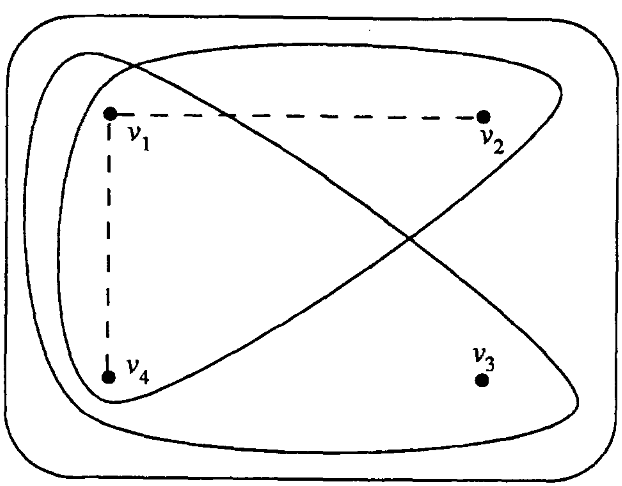

Fig. 7. Co-occurrence hypergraph for Example 4.

(B_4) has 4 complicating variables. It may be written:

$$\text{Problem } (B4)\begin{cases} \min x_1y_5 + y_6 + x_3y_5 + x_3 \\ \text{subject to:} \\ x_3y_6 - 25 \geq 0 \\ x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4 - 40 = 0 \\ y_5 = x_1y_4 \\ y_6 = x_2y_5 \\ x_1 = y_1 \\ x_2 = y_2 \\ x_3 = y_3 \\ x_4 = y_4 \\ \text{and the corresponding range constraints.} \end{cases}$$

Considering now complicating variables yields stronger results. Indeed a lower bound generalizing the lower bound of Theorem 2 can be obtained.

THEOREM 3. *Let (P) be a polynomial program with co-occurrence hypergraph H with 2-section H_2. Then the number of complicating variables in any reduction of (P) to a bilinear program (B) is at least*

$$n - \alpha(H_2)$$

where $\alpha(H_2)$ is the stability number of H_2.

Proof. Using a similar reasoning than in Proposition 1, at least $k-1$ complicating variables in (B) are associated with any product of k variables in (P). Moreover, these complicating variables must either be variables of that product or additional variables associated, directly or indirectly, with variables of that product only. So the number of complicating variables is not less than the number of vertices in a set containing $|E_i| - 1$ vertices from each edge of H. This number is bounded in turn by $n - \alpha(H_2)$. ■

Again, the bound is not always sharp but is reached for some problems. The 2-section H_2 of H for Example 5 is equal to the complete graph on four vertices with loops at all of them. So $\alpha(H_2) = 0$ and the reduction given above is narrow.

4.2. HYPERBOLIC PROGRAMS

We call *hyperbolic programs* those programs in which the objective function and left-hand sides of the constraints can be expressed, possibly after reduction to common denominators, as ratios of polynomials. If the denominators in the constraints are non-negative or non-positive, these constraints can be expressed as polynomials after multiplication by the denominator, but the sign of these denominators may be undetermined or unknown. Hyperbolic programs comprise all programs in which the objective and left-hand sides of constraints are obtained from constants and variables by operations of addition, subtraction, multiplication and division. These expressions need not be given as ratios of polynomials and reduction to such a form may be cumbersome. So ratios within larger expressions may first be removed. This is done by introducing a new variable equal to the inverse of the denominator, in a straightforward way when the sign of the latter is determined.

EXAMPLE 6. (Modification of the gravel box problem of Duffin, Peterson and Zener [17], Gochet and Smeers [27]).

$$\text{Problem } (H_1) \begin{cases} \min \dfrac{40}{x_1x_2x_3} + 20x_2x_3 + 40x_1x_2 + 10x_1x_3 \\ \text{subject to:} \\ -x_1 - x_2 - x_3 + 8 \leq 0 \\ x_1, x_2, x_3 > 0 \,. \end{cases}$$

Setting $y_1 = 1/(x_1x_2x_3)$, introducing two additional variables $y_2 = y_1x_1$ and $y_3 = y_2x_2$ to reduce the product $y_1x_1x_2x_3$ and one additional variable $x_3' = x_3$ to reduce the set of terms x_2x_3, x_1x_2 and x_1x_3, (which correspond to an odd cycle of G) yields a short and narrow reduction. The resulting bilinear program (B_5) has 3 complicating variables, i.e., x_1, y_2 and x_3. It may be written after renaming some of the variables $(x_2 \leftarrow y_2;\ y_2 \leftarrow x_2;\ x_3' \leftarrow y_4)$:

$$\text{Problem } (B_5) \begin{cases} \min 40y_1 + 20x_3y_2 + 40x_1y_2 + 10x_1y_4 \\ \text{subject to} \\ -x_1 - y_2 - x_3 + 8 \leq 0 \\ x_2 = x_1y_1 \\ y_3 = x_2y_2 \\ x_3y_3 = 1 \\ x_3 = y_4 \end{cases}$$

all variables being strictly positive.

Another class of programs in which a single additional and complicating variable is needed, as in the multiplicative programs discussed above, are the *linear hyperbolic programs* (e.g., Avriel *et al.* [4]). In such programs, a ratio of non-negative linear functions is to be minimized subject to linear constraints. Under the mild restriction that the feasible set does not reduce to a vector giving the value 0 to the denominator of the objective function (which can be easily checked) the above transformation applies in a straightforward way. (Note that problems in which the ratio is to be maximized can be reduced to the minimization case by exchanging the numerator and the denominator).

If the sign of the denominator is unknown and a value of 0 cannot be ruled out a priori, reduction is more difficult. In addition to setting $y_1 = 1/[h(x)]$ where the sign of $h(x)$ is unknown, one must add the pair of disjunctive constraints $h(x) \geqslant \varepsilon$ or $h(x) \leqslant -\varepsilon$ where ε is a small constant. The latter can be reduced to usual constraints involving a 0–1 variable by standard techniques, e.g., by setting $(2y_1 - 1)h(x) \leqslant \varepsilon$ where y_1 is a 0–1 variable. Reduction of 0–1 variables is discussed below. Various forms of *generalized* and of *nonlinear fractional programs* (Bernard and Ferland [12], Avriel *et al.* [4]) can be reduced to bilinear programs by combining standard techniques to express the minimum of a set of functions (to be maximized) with those given above.

4.3. FRACTIONAL EXPONENTS, TRIGONOMETRIC AND TRANSCENDENTAL FUNCTIONS, INTEGER AND 0–1 VARIABLES

Several further classes of programs can be reduced to polynomial and hence to bilinear programs. Problems with variables having fractional exponents, e.g., *signomial geometric programs* (Avriel and Williams [5]) can be reduced noting that $y = x^{p/q}$ where p and q are pairwise prime is equivalent to $x^p = y^q$ (such a reduction is used implicitly in an example of Floudas, Aggarwal and Ciric [21]) and then applying techniques described above. Note that each variable with fractional exponent is associated with 2 complicating variables.

EXAMPLE 7. (Stephanopoulos and Westerberg [43], Floudas, Aggarwal and Ciric [21]).

$$\text{Problem } (G_1) \begin{cases} \min x_1^{0.6} + x_2^{0.6} - 6x_1 - 4x_3 - 3x_4 \\ \text{subject to:} \\ -3x_1 + x_2 - 3x_3 = 0 \\ x_1 + 2x_3 \leqslant 4 \\ x_2 + 2x_4 \leqslant 4 \\ x_1 \leqslant 3 \\ x_3 \leqslant 1 \\ x_1, x_2, x_3, x_4 \geqslant 0 \end{cases}$$

The only nonlinear terms are the first two in the objective function. They are replaced by additional variables y_1 and y_2; then the constraints $x_1^3 = y_1^5$ and $x_2^3 = y_2^5$ are reduced to bilinear ones using 12 more additional variables. The resulting bilinear program (B_6) has 4 complicating variables, i.e., x_1, x_2, y_1 and y_2. After renaming some of the variables ($y_1 \leftarrow x_3$; $y_2 \leftarrow x_4$; $x_3 \leftarrow y_{13}$; $x_4 \leftarrow y_{14}$), it may be written:

$$\text{Problem } (B_6)\begin{cases} \min x_3 + x_4 - 6x_1 - 4y_{13} - 3y_{14} \\ \text{subject to:} \\ y_1 = x_1 \\ y_2 = x_1 y_1 \\ y_3 = x_3 \\ y_4 = x_3 y_3 \\ y_5 = x_3 y_4 \\ y_6 = x_3 y_5 \\ x_1 y_2 = x_3 y_6 \\ y_7 = x_2 \\ y_8 = x_2 y_7 \\ y_9 = x_4 \\ y_{10} = x_4 y_4 \\ y_{11} = x_4 y_{10} \\ y_{12} = x_4 y_{11} \\ x_2 y_8 = x_4 y_{12} \\ -3x_1 + x_2 - 3y_{13} = 0 \\ x_1 + 2y_{13} \leqslant 4 \\ x_2 + 2y_{14} \leqslant 4 \\ x_1 \leqslant 3 \\ y_{13} \leqslant 1\,, \end{cases}$$

all variables being non-negative.

Provided convergence conditions are satisfied, trigonometric and transcendental functions can, be approximated by their expansion in MacLaurin series. Bounded integer variables can be expanded in powers of 2 multiplied by binary variables. Binary variables x_j can be replaced by continuous ones, after adding the standard constraint $x_j(1 - x_j) = 0$ (Ragavachari [40]) which is equivalent to $x_j - x_j x_j' = 0$ and $x_j = x_j'$.

EXAMPLE 8. (Mladineo [33]).

$$\text{Problem } (G_2)\begin{cases} \min 4x_1 x_2 \sin(4\pi x_2) \\ \text{subject to:} \\ 0 \leqslant x_1 \leqslant 1 \\ 0 \leqslant x_2 \leqslant 1\,. \end{cases}$$

Expanding $\sin(4\pi x_2)$ as a MacLaurin series and keeping the first p terms allows to rewrite the objective function as

$$\min 4x_1 x_2 \sum_{k=1}^{p} \frac{(-1)^{k+1}(4\pi)^{2k-1}}{(2k-1)!}\, x_2^{2k-1} = \sum_{k=1}^{p} c_k x_1 x_2^{2k}$$

where

$$c_k = \frac{4(-1)^{k+1}(4\pi)^{2k-1}}{(2k-1)!} \quad \text{for} \quad k = 1, 2, \ldots, p .$$

Then using additional variables to substitute for powers of x_2 yields the bilinear program:

$$\text{Problem } (B_7) \begin{cases} \min \sum_{k=1}^{p} c_k x_1 y_{2k} \\ \text{subject to:} \\ y_1 = x_2 \\ y_j = y_{j-1} x_2 & \text{for } j = 2, 3, \ldots, 2p \\ 0 \leq x_j \leq 1 & \text{for } j = 1, 2 \\ 0 \leq y_i \leq 1 & \text{for } j = 1, 2, \ldots, 2p , \end{cases}$$

which has only 2 complicating variables.

It thus appears that a very large class of programs involving smooth functions can be reduced to bilinear programs, often in an exact way or with an approximation as precise as desired. They are thus amenable to solution, at least in principle. The numbers of additional and of complicating variables may, however, be large in some cases. Exact algorithms for finding short and narrow reductions are now available in the case of indefinite quadratic programs. Finding such reductions for more general cases appears to be difficult and is essentially an open problem.

Acknowledgements

Research of the first author has been supported in part by AFOSR grant 90-0008 to Rutgers University and in part by NSERG (Natural Sciences and Engineering Research Council of Canada) grant to Ecole des Hautes Etudes Commerciales, Montréal. Research of the second author has been supported by NSERG grant GP0036426 and FCAR (Fonds pour la Formation de Chercheurs et l'Aide à la Recherche) grants 89EQ4144 and 90NC0305, by a NSF Professorship for Women in Science at Princeton University, and by AFORS grant 90-0008 to Rutgers University.

References

1. Al-Khayyal, F. A. (1990), Generalized Bilinear Programming: Part I. Models, Applications, and Linear Programming Relaxation, Research Report, Georgia Institute of Technology.
2. Al-Khayyal, F. A. and J. E. Falk (1983), Jointly Constrained Biconvex Programming, *Mathematics of Operations Research* **8** (2), 273–286.
3. Al-Khayyal, F., R. Horst, and P. Pardalos (1991), Global Optimization of Concave Functions Subject to Separable Quadratic Constraints and of All-Quadratic Separable Problems, *Annals of Operations Research*.

4. Avriel, M., W. E. Diernert, S. Schaible, and I. Zang (1988), *Generalized Concavity*, New York: Plenum Press.
5. Avriel, M. and A. C. Williams (1971), An Extension of Geometric Programming with Applications in Engineering Optimization, *Journal of Engineering Mathematics* **5** (3), 187–194.
6. Balas, E. and C. S. Yu (1986), Finding a Maximum Clique in an Arbitrary Graph, *SIAM Journal on Computing* **15**, 1054–1068.
7. Baron, D. P. (1972), Quadratic Programming with Quadratic Constraints, *Naval Research Logistics Quarterly* **19**, 253–260.
8. Bartholomew-Biggs, M. C. (1976), A Numerical Comparison between Two Approaches to Nonlinear Programming Problems, *Technical Report #77*, Numerical Optimization Center, Hatfield, England.
9. Benacer, R. and T. Pham Dinh (1986), Global Maximization of a Nondefinite Quardatic Function over a Convex Polyhedron, pp. 65–76 in J.-B. Hirriart-Urruty (ed.), *Fermat Days 1985: Mathematics for Optimization*, Amsterdam: North-Holland.
10. Berge, C. (1983), *Graphes*, 3rd ed., Paris: Gauthier-Villars.
11. Berge, C. (1987), *Hypergraphes*, Paris, Gauthier-Villars.
12. Bernard, J. C. and J. A. Ferland (1989), Convergence of Interval-Type Algorithms for Generalized Fractional Programming, *Mathematical Programming* **43**, 349–363.
13. Carraghan, R. and P. M. Pardalos (1990), An Exact Algorithm for the Maximum Clique Problem, *Operations Research Letters* **9**, 375–382.
14. Colville, A. R. (1986), A Comparative Study on Nonlinear Programming Codes, *IBM Scientific Center Report 320-2949*, New York.
15. Dembo, R. S. (1972), Solution of Complementary Geometric Programming Problems, M.Sc. Thesis, Technion, Haifa.
16. Dembo, R. S. (1976), A Set of Geometric Programming Test Problems and Their Solutions, *Mathematical Programming* **10**, 192–213.
17. Duffin, R. J., E. L. Peterson, and C. Zener (1967), *Geometric Programming: Theory and Applications*, New York: Wiley.
18. Ecker, J. G. and R. D. Niemi (1975), A Dual Method for Quadratic Programs with Quadratic Constraints, *SIAM Journal on Applied Mathematics* **28**, 568–576.
19. Evans, D. H. (1963), Modular Design – A Special Case in Nonlinear Programming, *Operations Research* **11**, 637–647.
20. Flippo, O. E. (1989), *Stability, Duality and Decomposition in General Mathematical Programming*, Rotterdam: Erasmus University Press.
21. Floudas, C. A., A. Aggarwal, and A. R. Ciric (1989), Global Optimum Search for Nonconvex NLP and MINLP Problems, *Computers and Chemical Engineering* **13** (10), 1117–1132.
22. Floudas, C. A. and P. Pardalos (1990), *A Collection of Test Problems for Constrained Global Optimization*, Lecture Notes in Computer Science, 455, Berlin: Springer-Verlag.
23. Floudas, C. A. and V. Visweswaran (1990), A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs – I. Theory, *Computers and Chemical Engineering* **14** (12), 1397–1417.
24. Friden, C., A. Hertz, and D. de Werra (1990), Tabaris: An Exact Algorithm Based on Tabu Search for Finding a Maximum Independent Set in a Graph, *Computers and Operations Research* **17** (5), 437–445.
25. Garey, M. R., D. S. Johnson, and L. Stockmeyer (1976), Some Simplified NP-Complete Graph Problems, *Theoretical Computer Science* **1**, 237–267.
26. Geoffrion, A. M. (1972), Generalized Benders Decomposition, *Journal of Optimization Theory and Its Applications* **10**, 237–260.
27. Gochet, W. and Y. Smeers (1979), A Branch and Bound Method for Reversed Geometric Programming, *Operations Research* **27**, 982–996.
28. Hock, W. and K. Schittkowski (1981), *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems #187, Berlin: Springer-Verlag.
29. Horst, R. and H. Tuy (1990), *Global Optimization, Deterministic Approaches*, Berlin: Springer-Verlag.
30. Konno, H. (1976), Maximizing a Convex Quadratic Function Subject to Linear Constraints *Mathematical Programming* **11**, 117–127.

31. Konno, H. and T. Kuno (1989), Linear Multiplicative Programming, Preprint IHSS 89–13, Tokyo Institute of Technology.
32. Kough, P. F. (1979), The Indefinite Quadratic Programming Problem, *Operations Research* **27**, 516–533.
33. Mladineo, R. H. (1986), An Algorithm for Finding the Global Maximum of a Multimodal, Multivariate Function, *Mathematical Programming* **34**, 188–200.
34. Pardalos, P. M., J. H. Glick, and J. B. Rosen (1987), Global Minimization of Indefinite Quadratic Problems, *Computing* **39**, 281–291.
35. Pardalos, P. M. and J. B. Rosen (1986), Methods for Global Concave Minimization: A Bibliographic Survey, *SIAM Review* **28** (3), 367–379.
36. Pardalos, P. M. and J. B. Rosen (1987), *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science #268, Berlin: Springer Verlag.
37. Pham Dinh, T. and S. El Bernoussi (1989), Numerical Methods for Solving a Class of Global Nonconvex Optimization Problems, *International Series of Numerical Mathematics* **87**, 97–132.
38. Phan, H. (1982), Quadratically Constrained Quadratic Programming: Some Applications and a Method of Solution, *Zeitschrift für Operations Research* **26**, 105–119.
39. Phillips, A. T. and J. B. Rosen, (1990), Guaranteed ε-Approximate Solution to Indefinite Global Optimization, *Naval Research Logistics* **37**, 499–514.
40. Ragavachari, M. (1989), On Connections between Zero-One Integer Programming and Concave Programming under Linear Constraints, *Operations Research* **17**, 680–684.
41. Reeves, G. R. (1975), Global Minimization in Nonconvex All-Quadratic Programming, *Management Science* **22**, 76–86.
42. Sherali, H. and A. Alameddine (1990), A New Reformulation-Linearization Technique for Bilinear Programming Problems, Research Report, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute.
43. Stephanopoulos, G. and A. W. Westerberg (1975), The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization, *Journal of Optimization Theory and Applications* **15**, 285–309.
44. Simões, L. M. C. (1987), Search for the Global Optimum of Least Volume Trusses, *Engineering Optimization* **11**, 49–67.
45. Thoai, N. V. (1990), Application of Decomposition Techniques in Global Optimization to the Convex Multiplicative Programming Problem, Paper presented at the *Second Workshop on Global Optimization*, Sopron, Hungary.
46. Tuy, H. (1986), A General Deterministic Approach to Global Optimization via d.-c. Programming, pp. 137–162, in J. B. Hirriart-Urruty (ed.), *Fermat Days 1985: Mathematics for Optimization*, Amsterdam: North-Holland.
47. Visweswaran, V. and C. A. Floudas (1990), A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs – II. Application of Theory and Test Problems, *Computers and Chemical Engineering* **14** (12), 1419–1434.
48. Wolsey, L. A. (1981), A Resource Decomposition Algorithm for General Mathematical Programs, *Mathematical Programming Study* **14**, 244–257.