Integer Programming Subject to Monomial Constraints*

Christoph Buchheim[1], Dennis Michaels[2], Robert Weismantel[2]

[1] Zentrum Mathematik, Technische Universität München, Germany.
buchheim@ma.tum.de
[2] Institut für Mathematische Optimierung,
Otto-von-Guericke Universität Magdeburg, Germany
michaels@mail.math.uni-magdeburg.de weismant@mail.math.uni-magdeburg.de

Abstract. We investigate integer programs containing monomial constraints of the type $\prod_{i\in I} x_i^{\alpha_i} = b$. Due to the number-theoretic nature of these constraints, standard methods based on linear algebra cannot be applied directly. Instead, we present a reformulation resulting in integer programs with linear constraints and polynomial objective functions, using prime decompositions of the right hand sides b. Moreover, we show that minimizing a linear objective function with nonnegative coefficients over bivariate constraints is possible in polynomial time.

1 Introduction

Let $\mathbb{Z}_+$ denote the set of nonnegative integers. We consider integer programs of the form

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & \textstyle\prod_{i=1}^n x_i^{\alpha_{i,j}} = b_j \qquad \text{for } j = 1,\dots,m \\ & x \in \mathbb{Z}_+^n, \end{aligned} \tag{1}$$

where $c \in \mathbb{Z}^n$, $b \in (\mathbb{Z}_+ \setminus \{0\})^m$, and $\alpha \in \mathbb{Z}_+^{n\times m}$. We assume throughout that every variable appears in at least one of the monomial constraints, i.e., that for every $i \in \{1,\dots,n\}$ at least one $\alpha_{i,j}$ is non-zero. This assumption can be made without loss of generality, since a variable, i say, with $\alpha_{i,j} = 0$ for all j can always be set to zero if $c_i \le 0$. Otherwise, if $c_i > 0$, then the corresponding program is unbounded.

The number-theoretic nature of monomial constraints on integer variables explains why such constraints are difficult to handle by standard techniques based on linear algebra. Indeed, a notorious problem related to such constraints is the fact that the convex hull of integer feasible solutions in general contains integer infeasible points.

* The first author is partially supported by the German Science Foundation (DFG) under contract BU 2313/1-1. The second author is supported by the BMBF through the FORSYS-Partnerproject "TcellTalk".

For an example consider the constraint $xy = p$ with p prime, where x and y are nonnegative integer variables. This equation has only two feasible solutions, namely $(x, y) = (1, p)$ and $(x, y) = (p, 1)$. The convex hull of those two points, however, contains $p - 2$ additional, infeasible integer vectors,

$$(q, p + 1 - q) \in \mathbb{Z}^2_+, \text{ for } q = 2, \ldots, p - 1 \ ,$$

see Fig. 1(a). If p is not prime, the convex hull may even be full-dimensional and contain integer infeasible points in its interior, see Fig. 1(b).

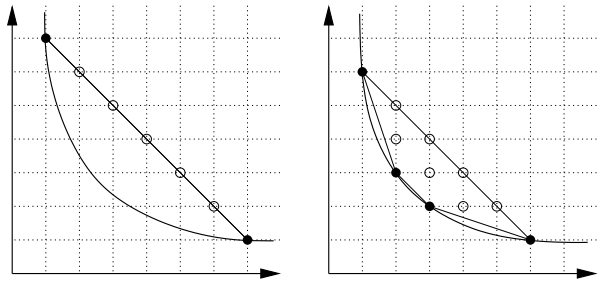

Fig. 1. The convex hull of feasible solutions for (a) $xy = 7$ and (b) $xy = 6$.

When intersecting this convex hull with other hyperplanes, e.g., with $x = y$, then one could even end up with an integer infeasible vertex. This fact limits on the one hand the applicability of traditional polyhedral techniques for nonlinear integer programs. On the other hand, this fact motivates the search for alternative approaches to tackle such a nonlinear integer problem.

It is our main objective to present such an alternative approach. We propose a transformation of Problem (1) into an integer *linear* program with polynomial objective function. For every monomial constraint

$$\prod_{i=1}^{n} x_i^{\alpha_{i,j}} = b_j \ ,$$

the new formulation explicitely models the possible distributions of prime factors of the right hand sides b_j to the variables x_i appearing on the left hand side. We show that such a reformulation is always possible in quadratic time, and we use this reformulation to show that the bivariate version of Problem (1) can be solved in polynomial time for non-negative costs c. Moreover, we present a further reformulation of the problem as a constrained quadratic binary optimization problem. This construction is polynomial in the case that the right hand sides b_j are polynomially bounded in the input length.

We call Problem (1) an integer monomial program. Such a program generalizes the set partitioning problem. This connection can be used to settle the complexity of the feasibility version of Problem (1).

Theorem 1. *It is strongly NP-complete to decide whether Problem* (1) *has a feasible solution, even if* $\alpha_{i,j} \leq 1$ *for all* i, j *and* $\sum_{i=1}^{n} \alpha_{i,j} \leq 3$ *for all* j.

Proof. We polynomially transform the set partitioning problem to Problem (1). For this, let $I_j \subseteq \{1, \ldots, n\}$, $j = 1, \ldots, m$, be given. Then, we ask whether there is a binary vector $z \in \{0,1\}^n$ such that

$$\sum_{i \in I_j} z_i = 1 \tag{2}$$

for $j = 1, \ldots, m$. We claim that Equation (2) holds true if and only if

$$\prod_{i \in I_j} (z_i + 1) = 2 \, .$$

Indeed, for $x \in \mathbb{Z}_+^n$, the monomial equation $\prod_{i \in I_j} x_i = 2$ is satisfied if and only if all $x_i \in \{1, 2\}$ for $i \in I_j$ and $\sum_{i \in I_j} (x_i - 1) = 1$. We conclude that the given set partitioning instance is feasible if and only if

$$\begin{aligned} \textstyle\prod_{i=1}^{n} x_i^{\alpha_{i,j}} &= 2 \quad \text{for } j = 1, \ldots, m \\ x &\in \mathbb{Z}_+^n \end{aligned}$$

has a feasible solution, where

$$\alpha_{i,j} = \begin{cases} 1 & \text{if } i \in I_j \\ 0 & \text{otherwise.} \end{cases}$$

As the feasibility version of the set partitioning problem is NP-complete even if all equations have a support of cardinality at most three [3], deciding whether Problem (1) has a feasible solution is NP-complete, too. □

In the following sections we will investigate the complexity of special cases of monomial integer programming problems. In doing so, we need some care in defining the encoding of the input. It will turn out that a central element of the analysis provided in the paper is based on prime decompositions of the right hand sides b_j. For a general integer b, a decomposition into its prime factors $p_1, \ldots, p_t$ can be computed in polynomial time in the encoding length $\langle b \rangle$ of b using a quantum computer model [6]. In the Turing computer model it is not known whether a prime factorization can be determined in polynomial time. The best known algorithms for the factorization of integers are sub-exponential (e.g., the general number field sieve algorithm [4]), but not polynomial. This motivates to include the prime decomposition of the right hand sides b_j into the input of Problem (1). We thus input, in binary encoding,

- a positive number n
- a positive number m
- for every $j = 1, \ldots, m$, a positive integer b_j, together with its unique prime decomposition, given by the pairs $(p_k, \mu_{k,j})$ with $b_j = \prod_k p_k^{\mu_{k,j}}$,

– for every $i = 1, \ldots, n$, an integer c_i,
– for every $j = 1, \ldots, m$ and $i = 1, \ldots, n$, a non-negative integer $\alpha_{i,j}$, such that $\sum_{j=1}^{m} \alpha_{i,j} \geq 1$ for all $i = 1, \ldots, n$

The objective is to solve Problem (1). Note that every $\mu_{k,j}$ is logarithmically bounded in b_j, as $p_k \geq 2$, and hence linear in the input size.

The paper is organized as follows: Section 2 introduces a reformulation of Problem (1) as a polynomial optimization problem subject to linear integer constraints. Section 3 is devoted to the investigation of the special bivariate case in which at most two variables appear in any constraint. In Section 4, we propose a binary quadratic programming model for Problem (1) which is of polynomial size if all right hand sides b_i are polynomially bounded.

2 A transformation based on prime decompositions

An important difficulty in tackling nonlinear integer programs stems from the fact that the convex hull of feasible solutions can contain infeasible integer points. The purpose of this section is to propose a reformulation of Problem (1) that turns all nonlinear constraints into linear ones in an extended space whose dimension is quadratic in the encoding length of the original problem input. The core of this reformulation is a transformation based on the prime factorization of the right hand sides. We consider the set of monomial constraints

$$\begin{aligned} \prod_{i=1}^{n} x_i^{\alpha_{i,j}} = b_j = \prod_{k=1}^{t} p_k^{\mu_{k,j}}, \quad \forall\, j = 1, \ldots, m, \\ x \in \mathbb{Z}_+^n \,. \end{aligned} \tag{3}$$

Let $\gamma_{i,k} := \min\{\lfloor \mu_{k,j}/\alpha_{i,j} \rfloor \mid j = 1, \ldots, m \text{ with } \alpha_{i,j} \geq 1\}$. We now associate with each variable x_i new integer variables $y_{i,k} \in \{0, 1, \ldots, \gamma_{i,k}\}$, for $k = 1, \ldots, t$, which count how often p_k divides the value of variable x_i, and consider the following integer linear system:

$$\begin{aligned} \textstyle\sum_{i=1}^{n} \alpha_{i,j} y_{i,k} = \mu_{k,j}, \qquad & \forall\, k = 1, \ldots, t,\ \forall\, j = 1, \ldots, m, \\ y_{i,k} \in \{0, \ldots, \gamma_{i,k}\}, & \forall\, i = 1, \ldots, n,\ \forall k = 1, \ldots, t. \end{aligned} \tag{4}$$

Proposition 1. *Each feasible solution to system* (3) *corresponds to a feasible solution of the linear system* (4)*, and vice versa.*

Proof. Let $x \in \mathbb{Z}_+^n$ be feasible for (3). Since $\sum_{j=1}^{m} \alpha_{i,j} \geq 1$ for all $i = 1, \ldots, n$, each $x_i \in \mathbb{Z}_+$ divides at least one b_j. It follows that

$$x_i \quad = \quad \prod_{k=1}^{t} p_k^{w_{i,k}}, \quad \text{for suitable } w_{i,k} \in \mathbb{Z}_+.$$

As $x_i^{\alpha_{i,j}}$ must be a divisor of b_j, we conclude that $w_{i,k} \in \{0, 1, \ldots, \gamma_{i,k}\}$. Now setting $y_{i,k} = w_{i,k}$, for all i and k, we obtain for all $j = 1, \ldots, m$ that

$$\prod_{k=1}^{t} p_k^{\mu_{k,j}} = b_j \;=\; \prod_{i=1}^{n} x_i^{\alpha_{i,j}} = \prod_{i=1}^{n}\prod_{k=1}^{t} p_k^{\alpha_{i,j}y_{i,k}} = \prod_{k=1}^{t} p_k^{\sum_{i=1}^{n}\alpha_{i,j}y_{i,k}}.$$

Comparing exponents and using the uniqueness of the prime decomposition, it follows that for all $j = 1, \ldots, m$

$$\sum_{i=1}^{n} \alpha_{i,j} y_{i,k} = \mu_{k,j}, \quad \text{for all } k = 1, \ldots, t.$$

Thus $y_{i,k}$ is feasible for the constraints in (4).

Now, assume that $y_{i,k} \in \{0, \ldots, \gamma_{i,k}\}$ is a feasible solution for system (4). Then, for each $j = 1, \ldots, m$, we have that

$$p_k^{\mu_{k,j}} \;=\; p_k^{\sum_{i=1}^{n}\alpha_{i,j}y_{i,k}}, \quad \text{for all } k = 1, \ldots, t.$$

We define $x_i = \prod_{k=1}^{t} p_k^{y_{i,k}}$, for all $i = 1, \ldots, n$. This implies

$$b_j = \prod_{k=1}^{t} p_k^{\mu_{k,j}} \;=\; \prod_{k=1}^{t} p_k^{\sum_{i=1}^{n}\alpha_{i,j}y_{i,k}} = \prod_{i=1}^{n}\Big(\prod_{k=1}^{t} p_k^{y_{i,k}}\Big)^{\alpha_{i,j}} = \prod_{i=1}^{n} x_i^{\alpha_{i,j}},$$

for all $j \in \{1, \ldots, m\}$. □

In order to complete the reformulation of Problem (1), we still need to transform the original objective function to the new model. From a solution $y_{i,k}$ of system (4) we can reconstruct a feasible solution for the variables x_i using

$$x_i \quad = \quad \prod_{k=1}^{t} p_k^{y_{i,k}}.$$

Then, the objective function $c^\top x$ turns into

$$c^\top x = \sum_{i=1}^{n} c_i \prod_{k=1}^{t} p_k^{y_{i,k}} \quad =: \quad f(y). \tag{5}$$

Unfortunately, $f(y)$ is a highly nonlinear function so that no tools are available to solve the transformed system max $f(y)$ subject to (4). With the help of additional binary variables, we claim that we can write down a polynomially sized linear system and polynomial objective function that model problem (1) correctly.

Theorem 2. *In quadratic time, Problem* (1) *can be transformed to a linear integer problem with polynomial objective function.*

Proof. From Proposition 1, it follows that Problem (1) is equivalent to maximizing function $f(y)$ of Formula (5) over the linear equation system stated in (4). Now, we replace each integer variable $y_{i,k}$ by the sum of $\gamma_{i,k}$ binary variables $z^r_{i,k}$, i.e. $y_{i,k} = \sum_{r=1}^{\gamma_{i,k}} z^r_{i,k}$. Then, $f(y)$ reads as

$$\sum_{i=1}^{n} c_i \prod_{k=1}^{t} p_k^{\sum_{r=1}^{\gamma_{i,k}} z^r_{i,k}} = \sum_{i=1}^{n} c_i \prod_{k=1}^{t} \prod_{r=1}^{\gamma_{i,k}} \left(1 + (p_k - 1) z^r_{i,k}\right) \quad =: \quad g(z). \tag{6}$$

In summary, Problem (1) is equivalent to

$$\begin{aligned} \max \qquad & g(z) \\ & \textstyle\sum_{i=1}^{n} \alpha_{i,j} \sum_{r=1}^{\gamma_{i,k}} z^r_{i,k} = \mu_{k,j}, \quad \forall\, k, j, \\ & z^r_{i,k} \in \{0,1\}, \forall\, k, i, r. \end{aligned} \tag{7}$$

This model contains at most one binary variable $z^r_{i,k}$ for each variable x_i and each prime factor p_k appearing in some right hand side b_j with $\alpha_{i,j} \geq 1$, where we distinguish between different appearances of the same prime factor in a given right hand side. It follows that the total number of binary variables is bounded by $n(\sum_{j=1}^{m} \log b_j)$, which is quadratic in the input length. □

Note that the presented construction is not linear in general. For example, to transform the single monomial constraint $\prod_{i=1}^{n} x_i = b$ into a linear one, one needs $n \log b$ variables $y_{i,k}$. The reason is that every prime factor of b can appear in every variable x_i. However, it is easy to see that the construction becomes linear if the degree of all monomials is bounded by a constant.

In the case where $\mu_{i,k} \leq 1$ for all i, k, i.e., where all right hand sides are square-free, the variables $z^r_{i,k}$ are not needed, and the system (7) specializes to the easier problem

$$\begin{aligned} \max \textstyle\sum_{i=1}^{n} c_i \prod_{k=1}^{t} \left(1 + (p_k - 1) y_{i,k}\right) \\ \textstyle\sum_{i=1}^{n} \alpha_{i,j} y_{i,k} = \mu_{k,j}, \quad \forall\, k, j, \\ y_{i,k} \in \{0,1\}, \forall\, k, i. \end{aligned}$$

We remark that Theorem 2 cannot be proven using the standard linearization approach in which every variable x_i is expanded $x_i = \sum_l 2^l y_{il}$ with $y_{i,l} \in \{0,1\}$. Indeed, this approach would require to introduce additional binary variables in order to linearize the monomial constraints. Since the numbers $\alpha_{i,j}$ are part of the input and not constant, the number of such extra linearization variables grows exponentially with the size of the input of Problem (1).

Example 1. For pairwise distinct prime numbers p_1, p_2, p_3, consider the following instance of Problem (1):

$$\begin{array}{lrrrrrcl}
\min & \sum_{i=1}^{15} x_i & & & & & & \\
\text{s.t.} & x_1 & x_2 & x_3 & x_4 & x_5 & = & b_1 := p_1^4\, p_2^2\, p_3^1, \\
 & x_6 & x_7 & x_8 & x_9 & x_{10} & = & b_2 := p_1^4\, p_2^2\, p_3^1, \\
 & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & = & b_3 := p_1^4\, p_2^2\, p_3^1, \\
 & & & x_1 & x_6 & x_{11} & = & b_4 := p_1^2\, p_2^4\, p_3^0, \\
 & & & x_2 & x_7 & x_{12} & = & b_5 := p_1^3\, p_2^2\, p_3^0, \\
 & & & x_3 & x_8 & x_{13} & = & b_6 := p_1^3\, p_2^0\, p_3^1, \\
 & & & x_4 & x_9 & x_{14} & = & b_7 := p_1^4\, p_2^1\, p_3^2, \\
 & & & x_5 & x_{10} & x_{15} & = & b_8 := p_1^1\, p_2^1\, p_3^1, \\
 & & & & & x & \in & \mathbb{Z}_+^{15}\,.
\end{array} \tag{8}$$

By taking the product of the first three constraints and the product of the last five constraints, we obtain that each feasible solution of Problem (8) must satisfy both $\prod_{i=1}^{15} x_i = p_1^{12}\ p_2^6\ p_3^3$ and $\prod_{i=1}^{15} x_i = p_1^{13}\ p_2^8\ p_3^4$. This shows that Problem (8) is infeasible. We have created several test instances of type (8) characterized in Table 1. The second column contains the values chosen for the prime numbers p_1, p_2 and p_3. In column $b_{\max}$, the largest right-hand-side that occurs in the test instance is listed, while column $u_{\max}$ denotes the maximum upper bound induced on some variable x_i. For solving the test instances we

name	$p_1/p_2/p_3$	$b_{\max}$	$u_{\max}$	time (sec)	BaR It.
p01	2/3/5	1200	720	102.14	10965
p02	2/3/7	2952	1008	110.19	10803
p03	2/3/11	5808	1584	148.73	14681
p04	2/3/13	8112	1872	139.96	15323
p05	3/5/2	5625	4050	270.14	28305
p06	3/5/7	19845	14175	777.07	74961
p07	3/5/11	49005	22275	975.23	98239
p08	3/5/13	68445	26325	862.08	88427
p09*	5/7/2	61250	60025	1829.96	194227
p10*	5/7/3	91875	60025	2558.92	259655
p11*	5/7/11	529375	336875	3956.05	396555
p12*	5/7/13	739375	398125	2068.02	509757

Table 1. Characteristics and computational results for test instances of Problem (8). For instances p09–p12 marked with $\star$, BARON was not able to guarantee infeasibility as bounds on variables were too wide.

used the software package BARON [7], a state-of-the-art global solver for mixed-integer nonlinear optimization problems. All computations have been carried out in a GAMS 22.5 environment [5] using BARON v. 7. 8. 1 (`epsr`$= 1.00E-09$,

epsa= $1.00E-09$, isotol= $1.00E-04$, nlpsol=Snopt, lpsol=Cplex) on a 3GHz Dual-Core AMD Opteron(tm) Processor 8222 SE with 64GB Ram. The solution time and the number of BARON iterations (Bar It.) needed to prove infeasibility are also reported in Table 1. The computational results indicate that proving infeasibility of the test instances in their standard formulation (8) strongly depends on the size of the right-hand-sides and of the upper bounds on the variables, though the underlying structure, the core of the problem, is identical for all test instances p01–p12.

To see this, let us consider the reformulation of problem (8) given by

$$\begin{array}{l} \min \sum_{i=1}^{15} \; p_1^{y_{i,1}} \; p_2^{y_{i,2}} \; p_3^{y_{i,3}} \\ \text{s.t.} \; \sum_{i=1}^{15} \alpha_{i,j} y_{i,k} = \mu_{j,k}, \quad k \in \{1,2,3\}, \; j \in \{1,\ldots,8\}, \end{array} \tag{9}$$

where $\alpha_{i,j} \in \{0,1\}$ reflects the exponent of variable x_i in constraint j and $\mu_{j,k}$ is the multiplicity of p_k in b_j. Using the integer reformulation (9), infeasibility could be proven by BARON in the pre-processing step with less than 0.02 sec for all test instances p01–p12. This example shows that the reformulation suggested by Theorem 2 can capture the combinatorial structure of the problem much better than the original formulation. □

Based on Proposition 1, there is little hope in solving Problem (7) efficiently, in general. Interestingly, if we impose additional structure on the constraints, polynomial time algorithms are available to tackle the corresponding feasibility and optimization questions. This topic is discussed in the next section.

3 Bivariate constraints

The set partitioning problem turns easy as soon as every constraint has a support bounded by two. In our context, this translates to the fact that Problem (1) is easy if all monomial degrees and all right hand sides are bounded by two. In this section we show a much more general result. We consider the case where Problem (1) only involves bivariate monomials, but do not bound the right hand sides. In this case, it is appropriate to rewrite Problem (1) as

$$\begin{array}{ll} \max & c^\top x \\ \text{s.t.} & x_i^{\alpha_{i,j}} x_j^{\beta_{i,j}} = b_{i,j} = \prod_{k=1}^{t} p_k^{\mu_k(i,j)} \quad \text{for } (i,j) \in I, \; i<j, \\ & x \in \mathbb{Z}_+^n, \end{array} \tag{10}$$

where the index set $I \subseteq \{1,\ldots,m\}^2$ is given and we can assume $\alpha_{i,j}, \beta_{i,j} \geq 1$. It is the topic of this section to show that variants of this problem can be solved efficiently. In the following, we consider the undirected graph $G = (V, E)$ with $V = \{1,\ldots,n\}$ and $(i,j) \in E$ if and only if $(i,j) \in I$ and $i < j$, or $(j,i) \in I$ and $j < i$.

Proposition 2. *If every non-trivial component of G contains an edge (i,j) such that b_{ij} is polynomially bounded in the input size, then Problem* (10) *can be solved in polynomial time.*

Proof. We may assume that the graph G is connected, as otherwise the problem decomposes. Let $b_{i,j}$ be polynomial in the input size for some $(i,j) \in I$. The constraint $x_i^{\alpha_{i,j}} x_j^{\beta_{i,j}} = b_{i,j}$ has at most one solution for every subset of the set of prime factors of $b_{i,j}$. Since $b_{i,j}$ can have at most $\log b_{i,j}$ many prime factors, the number of solutions is thus bounded by $b_{i,j}$. As G is connected, fixing the value of x_i either implicitly fixes all variables or leads to a contradiction. This can be checked by a depth first traversal of G. □

The proof of Proposition 2 does not work for monomials of higher degree. In fact, a crucial advantage in the bivariate case is that fixing a single variable in a connected component of G fixes all other variables in this component. This fact is also used in the following proof.

Theorem 3. *It can be checked in polynomial time whether Problem* (10) *has a solution or is infeasible. Moreover, Problem* (10) *can be solved in polynomial time if* $c \geq 0$.

Proof. Again, we may assume that the graph G is connected. By Theorem 2, we have to solve the problem

$$\begin{aligned} \max \quad & \textstyle\sum_{i=1}^{n} c_i \prod_{k=1}^{t} p_k^{y_{i,k}} \\ \text{s.t.} \quad & \alpha_{i,j} y_{i,k} + \beta_{i,j} y_{j,k} = \mu_k(i,j) \qquad \forall k \; \forall\, (i,j) \in I, \text{ with } i < j \\ & y_{i,k} \in \{0, \ldots, \gamma_{i,k}\} \;\; \forall k \; \forall i \; . \end{aligned} \tag{11}$$

Feasibility of (11) can be checked in the following way: Let $k \in \{1, \ldots, t\}$ be fixed, and choose a variable $y_{i,k}$. As each equation involves only two variables, fixing $y_{i,k}$ to one of its values either leads to a uniquely feasible integer solution for all variables $y_{i,k}$, $i \in J$, or leads to a non-solvable integer system in the remaining variables. This can obviously be checked in linear time. This means, testing solvability requires for each $k \in \{1, \ldots, t\}$ evaluating at most $\gamma_{i,k} + 1$ possibilities for $y_{i,k}$, and the total number of evaluations is polynomially bounded in the input size.

To determine an optimal solution for Problem (11), we first fix k to one of its values. Then, the set of equations

$$\alpha_{i,j} y_{i,k} + \beta_{i,j} y_{j,k} = \mu_k(i,j) \;\; \forall\, (i,j) \in I, \text{ with } i < j \text{ and } i,j \in J$$

has a polynomial number of feasible solutions. One can easily verify that these solutions are of the form

$$y_{i,k} = y'_{i,k} + \lambda_k \bar{y}_i, \quad \lambda_k \in \{0, \ldots, l_k\}$$

with $y'_{i,k} \in \mathbb{Z}$ for all i,k and $\bar{y}_i \in \mathbb{Z} \setminus \{0\}$ for all i. Note that $\bar{y}_i$ does not depend on k, but only on the $\alpha_{i,j}$ and $\beta_{i,j}$. Using this reformulation, Problem (11) can be rewritten as

$$\begin{aligned} \max \quad & \textstyle\sum_{i \in J} \bar{c}_i \Big(\prod_{k=1}^{t} p_k^{\lambda_k} \Big)^{\bar{y}_i} \\ \text{s.t.} \quad & \lambda_k \in \{0, \ldots, l_k\} \;\; \forall k \; , \end{aligned} \tag{12}$$

where $\bar{c}_i = c_i \prod_{k=1}^{t} p_k^{y'_{i,k}} \geq 0$. Thus, the function

$$f\colon (0,\infty) \to \mathbb{R}, \quad f(x) = \sum_{i \in J} \bar{c}_i x^{\bar{y}_i}$$

is convex. In particular, its maximum over the interval $[1, \prod_{k=1}^{t} p_k^{l_k}]$ is attained at one of the end points. In other words, an optimal solution for Problem (12) is either $\lambda_k = 0$ for all k or $\lambda_k = l_k$ for all k. □

Example 2. Consider the problem

$$\begin{aligned} \max \quad & 2x_1 + 3x_3 + x_4 \\ \text{s.t.} \quad & x_1^2 x_2 = 2^3 \cdot 3^4 \cdot 5^3 = 81.000 \\ & x_1 x_3 = 2^1 \cdot 3^3 \cdot 5^2 = 1.350 \\ & x_2 x_4^2 = 2^5 \cdot 3^4 \cdot 5^7 = 202.500.000 \\ & x_3 x_4 = 2^2 \cdot 3^3 \cdot 5^4 = 67.500 \\ & x \in \mathbb{Z}_+^4. \end{aligned}$$

Then $p_1 = 2$, $p_2 = 3$, and $p_3 = 5$. The resulting system of linear equations is

$$\begin{array}{lll} 2y_{1,1} + y_{2,1} = 3 & 2y_{1,2} + y_{2,2} = 4 & 2y_{1,3} + y_{2,3} = 3 \\ y_{1,1} + y_{3,1} = 1 & y_{1,2} + y_{3,2} = 3 & y_{1,3} + y_{3,3} = 2 \\ y_{2,1} + 2y_{4,1} = 5 & y_{2,2} + 2y_{4,2} = 4 & y_{2,3} + 2y_{4,3} = 7 \\ y_{3,1} + y_{4,1} = 2 & y_{3,2} + y_{4,2} = 3 & y_{3,3} + y_{4,3} = 4 \end{array}$$

Then

$$(y_{i,k})_{i,k} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 4 & 3 \\ 1 & 3 & 2 \\ 1 & 0 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix},$$

with upper bounds $l_1 = 1$, $l_2 = 2$, and $l_3 = 1$. The two candidate solutions are thus given by $\lambda = (0,0,0)$ and $\lambda = (1,2,1)$, they correspond to solutions

$$x_1 = (2^0 3^0 5^0, 2^3 3^4 5^3, 2^1 3^3 5^2, 2^1 3^0 5^2) = (1, 81.000, 1.350, 50)$$

and

$$x_2 = (2^1 3^2 5^1, 2^1 3^0 5^1, 2^0 3^1 5^1, 2^2 3^2 5^3) = (90, 10, 15, 4.500)\,.$$

The corresponding objective values are 4.102 and 4.725. □

It remains open whether Problem (10) can be solved in polynomial time for general objective functions. We conjecture that the problem is NP-hard. In fact, the problem becomes NP-hard if instead of the prime factorization of b we fix an arbitrary decomposition into factors that may be distributed to the variables on the left hand sides. This problem is NP-hard even in dimension two.

Problem 1. Given a set of integers $s_1, \ldots, s_n$, minimize $x_1 + x_2$ subject to

$$x_1 x_2 = \prod_{i=1}^{n} s_i$$

and $x_1 = \prod_{i \in I} s_i$ for some $I \subseteq \{1, \ldots, n\}$.

Theorem 4. *Problem 1 is NP-hard.*

Proof. To prove our claim, we will make use of the fact that the *subset product problem* is NP-complete [3], which is defined as follows:

(Q) Given a finite set $A = \{1, \ldots, n\}$, positive weights $s_a \in \mathbb{Z}_+$, $a \in A$, and a positive integer $B \in \mathbb{Z}_+$, is there a subset $A' \subseteq A$ such that $\prod_{a \in A'} s_a = B$?

Assume that there exists an algorithm that solves any instance of Problem 1 in polynomial time and that an arbitrary instance of the subset product problem **(Q)** is given. As for $B = 1$ problem **(Q)** can be easily solved we may assume that $B \geq 2$ and $s_a \geq 2$, $a \in A$.

We first introduce a new item $n+1$ and define its weight as $s_{n+1} := \frac{s_0}{B}$, where $s_0 := \prod_{a \in A} s_a$. Note that if B is not a divisor of s_0 then it follows that there cannot exist a subset $A' \subseteq A$ with $\prod_{a \in A'} s_a = B$. Therefore, we may assume that B divides s_0 implying that $s_{n+1} \in \mathbb{Z}_+$.

Now for fixed $a \in A$, we define a new item $n+2$ with weight $s_{n+2} := s_a B$ and consider the following optimization problem:

$$\begin{aligned} \nu_a := \ & \min x_1 + x_2 \\ & \text{s.t. } x_1 x_2 = \Big(\prod_{i \in A \setminus \{a\}} s_i \Big)\, s_{n+1}\, s_{n+2}, \\ & \qquad x_1 \in X := \Big\{ \prod_{i \in I} s_i \mid I \subseteq (A \setminus \{a\}) \cup \{n+1, n+2\} \Big\}. \end{aligned}$$

We have that $\big(\prod_{i \in A \setminus \{a\}} s_i\big) s_{n+1} s_{n+2} = \big(\prod_{i \in A \setminus \{a\}} s_i\big) \frac{s_0}{B} s_a B = s_0^2$. Thus, if we drop the condition $x_1 \in X$, then simple arguments from analysis show that ν_a equals $2s_0$ which can be only attained at (s_0, s_0). For the discrete case, we now claim that $\nu_a = 2s_0$ if and only if there exists a subset $A' \subseteq A \setminus \{a\}$ with $B = \prod_{i \in A'} s_i$. Clearly, if there exists an $A' \subseteq A \setminus \{a\}$ with $B = \prod_{i \in A'} s_i$, we can choose

$$x_1 = s_{n+1} \prod_{i \in A'} s_i = \frac{s_0}{B} B = s_0 \ ,$$

yielding $x_2 = s_0$ and $\nu_a = 2s_0$.

Now assume that $\nu_a = 2s_0$ with optimal solution $(x_1, x_2) = (s_0, s_0)$. Without loss of generality, s_{n+1} is assigned to x_1. Then s_{n+2} cannot be assigned to x_1, as otherwise $x_1 \geq s_{n+1} s_{n+2} = \frac{s_0}{B} s_a B > s_0$. Thus we have $x_1 = s_{n+1} w_1$ and $x_2 = s_{n+2} w_2$ with $w_1 w_2 = \prod_{i \in A \setminus \{a\}} s_i$. From the condition $x_1 \in X$ we derive

that $w_j \in \left\{ \prod_{i \in I} s_i \mid I \subseteq A \setminus \{a\} \right\}$ for $j = 1, 2$. From $x_1 = x_2 = s_0$, it moreover follows that

$$w_1 = \frac{s_0}{s_{n+1}} = s_0 \frac{B}{s_0} = B \quad \text{and} \quad w_2 = \frac{s_0}{s_{n+2}} = \frac{\prod_{i \in A \setminus \{a\}} s_i}{B} \; .$$

We can hence conclude that there must exist an $A' \subseteq A \setminus \{a\}$ with $B = \prod_{a \in A'} s_i$. This shows that for answering question **(Q)** it suffices to compute the value ν_a for all $a \in A$. Therefore, Problem 1 is NP-hard. □

4 Pseudopolynomial reduction to quadratic programming

The proof of Theorem 1 shows that Problem (1) remains NP-hard even if $b_j \leq 2$ for all j. In this section, we aim at a further transformation of Problem (1) under the assumption that all b_j are small. Our objective is to obtain an equivalent formulation that can be addressed by standard techniques for quadratic 0–1 programming. In the following, we consider the polytope P given as the convex hull of feasible solutions of Problem (1), i.e.,

$$P = \text{conv} \left\{ x \in \mathbb{Z}^n_+ \mid \prod_{i=1}^{n} x_i^{\alpha_{i,j}} = b_j \text{ for all } j = 1, \ldots, m \right\} \subseteq \mathbb{R}^n \; .$$

Problem (1) is equivalent to the maximization of an arbitrary linear objective function over P, so our aim is to derive tight linear relaxations of P.

For this, let P^* denote the convex hull of vectors $(x, y) \in \mathbb{Z}^n_+ \times \mathbb{Z}^{nt}_+$ satisfying

$$\begin{aligned} x_i &= \textstyle\prod_{k=1}^{t} p_k^{y_{i,k}}, \quad \forall\, i, \\ y_{i,k} &\in \{0, \ldots, \gamma_{i,k}\}, \forall\, i, k. \end{aligned}$$

By Proposition 1, an integer vector $x \in \mathbb{Z}^n_+$ belongs to P if and only if there is a vector $y \in \mathbb{Z}^{nt}_+$ that satisfies

$$\sum_{i=1}^{n} \alpha_{i,j} y_{i,k} = \mu_{k,j}, \; \forall k, j$$

such that $(x, y) \in P^*$. It is thus desirable to understand the polyhedral structure of P^*.

Theorem 5. *The polytope P^* is a projection of a face of a boolean quadric polytope Q^*. The latter can be constructed in time quadratic in the input length of Problem* (1) *plus b.*

Proof. First, we introduce z-variables as in the proof of Theorem 2, which is possible in quadratic time, yielding a polytope P^{**} given as the convex hull of feasible solutions of

$$\begin{aligned} x_i &= \textstyle\prod_{k=1}^{t} \prod_{r=1}^{\gamma_{i,k}} \left(1 + (p_k - 1) z^r_{i,k}\right), \; \forall\, i, \\ z^r_{i,k} &\in \{0, 1\}, \qquad\qquad\qquad\qquad \forall\, k, i, r. \end{aligned}$$

As for every $i \in \{1,\dots,n\}$ there is a $j \in \{1,\dots,m\}$ with $\alpha_{i,j} \geq 1$, we have

$$\sum_{k=1}^{t} \gamma_{i,k} \leq \sum_{k=1}^{t} \alpha_{i,j}\gamma_{i,k} \leq \sum_{k=1}^{t} \mu_{k,j} \leq \log b_j \ .$$

The number of subsets of $J_i = \{(k,r) \mid k = 1,\dots,t,\ r = 1,\dots,\gamma_{i,k}\}$ is thus bounded by $B = \max\{b_j \mid j = 1,\dots,m\}$. In particular, we can introduce new binary variables

$$z_{i,L} = \prod_{(k,r)\in L} z_{i,k}^r$$

for every i and every $L \subseteq J_i$. Then

$$x_i = \prod_{k=1}^{t} \prod_{r=1}^{\gamma_{i,k}} \left(1 + (p_k - 1)z_{i,k}^r\right) = \sum_{L\subseteq J_i} c_L z_{i,L}$$

with $c_L = \prod_{(k,r)\in L}(p_k - 1)$. Let P^{***} denote the convex hull of all feasible solutions of the new model, i.e.,

$$P^{***} = \operatorname{conv}\left\{(x,z) \in \mathbb{Z}_+^n \times \{0,1\}^{Bn} \mid x_i = \sum_{L\subseteq J_i} c_L z_{i,L}\right\} .$$

By construction, P^* is a projection of P^{***}, via

$$y_{i,k} = \sum_{r=1}^{\gamma_{i,k}} z_{i,k}^r = \sum_{r=1}^{\gamma_{i,k}} z_{i,\{(k,r)\}} \ .$$

Moreover, as each variable x_i is an affine combination of variables $z_{i,L}$, the polytope P^{***} is isomorphic to its own projection to the space of z-variables. The latter projection corresponds to the standard linearization of an unconstrained polynomial 0–1 optimization problem over the basic variables $z_{i,k}^r$, with a set of monomials $z_{i,L}$ that is closed under taking submonomials. By Corollary 3.4 in [1], it follows that P^{***} is isomorphic to a face of a boolean quadric polytope Q^*. Following the construction given in [2], the dimension of Q^* can be bounded by four times the dimension of P^{***}. As the latter is at most quadratic in the dimension of the original problem plus b, the result follows. □

Example 3. Consider the single monomial constraint $x_1^2 x_2 x_3^2 = 2^2 5^2 = 100$. Then P^* is defined as the convex hull of all vectors $(x,y) \in \mathbb{Z}_+^9$ satisfying

$$\begin{aligned} &x_1 = 2^{y_{1,1}}5^{y_{1,2}}, \quad x_2 = 2^{y_{2,1}}5^{y_{2,2}}, \quad x_3 = 2^{y_{3,1}}5^{y_{3,2}} \\ &y_{1,1}, y_{1,2}, y_{3,1}, y_{3,2} \in \{0,1\}, \quad y_{2,1}, y_{2,2} \in \{0,1,2\} \ , \end{aligned}$$

and P^{**} is spanned by the feasible solutions of

$$\begin{aligned} x_1 &= (z_{1,1}^1 + 1)(4z_{1,2}^1 + 1) \\ x_2 &= (z_{2,1}^1 + 1)(z_{2,1}^2 + 1)(4z_{2,2}^1 + 1)(4z_{2,2}^2 + 1) \\ x_3 &= (z_{3,1}^1 + 1)(4z_{3,2}^1 + 1) \\ z &\in \{0,1\}^8 \ . \end{aligned}$$

The polytope P^{***} obtained from multiplication and linearization is then defined over 21 binary variables $z_{i,L}$, corresponding to the monomials

$$\begin{array}{ll}
(\deg 1) & z^1_{1,1},\ z^1_{1,2},\ z^1_{2,1},\ z^2_{2,1},\ z^1_{2,2},\ z^2_{2,2},\ z^1_{3,1},\ z^1_{3,2} \\
(\deg 2) & z^1_{1,1}z^1_{1,2},\ z^1_{2,1}z^2_{2,1},\ z^1_{2,1}z^1_{2,2},\ z^1_{2,1}z^2_{2,2},\ z^2_{2,1}z^1_{2,2},\ z^2_{2,1}z^2_{2,2},\ z^1_{2,2}z^2_{2,2},\ z^1_{3,1}z^1_{3,2} \\
(\deg 3) & z^1_{2,1}z^2_{2,1}z^1_{2,2},\ z^1_{2,1}z^2_{2,1}z^2_{2,2},\ z^1_{2,1}z^1_{2,2}z^2_{2,2},\ z^2_{2,1}z^1_{2,2}z^2_{2,2} \\
(\deg 4) & z^1_{2,1}z^2_{2,1}z^1_{2,2}z^2_{2,2}\ .
\end{array}$$

It is isomorphic to a face of a boolean quadric polytope, corresponding to a quadratic function over 20 binary variables [2]. □

Theorem 5 shows that Problem (1) can be polynomially reformulated as a binary quadratic programming problem with additional linear constraints

$$\sum_{i=1}^{n} \alpha_{i,j} \sum_{r=1}^{\gamma_{i,k}} z^r_{i,k} = \mu_{k,j},\ \forall k, j$$

whenever the right hand sides of the monomial constraints are polynomially bounded in the input length. For binary quadratic programming problems, many well-studied and practically fast solution methods exist, based on integer or semidefinite programming techniques. Moreover, this approach remains feasible even when monomial constraints as in Problem (1) are combined with arbitrary linear constraints.

Theorem 6. *The polytope P is a projection of a face of a boolean quadric polytope Q. If all multiplicities $\mu_{k,j}$ are bounded by a constant, then Q can be constructed in polynomial time.*

Proof. By construction, the polytope P is a projection of the convex hull of integer points in the intersection of the polytope P^{***} constructed in the proof of Theorem 5 with the constraints

$$\sum_{i=1}^{n} \alpha_{i,j} \sum_{r=1}^{\gamma_{i,k}} z_{i,\{(k,r)\}} = \mu_{k,j} \tag{13}$$

for all k and j. We extend P^{***} by introducing further monomials over the same set of basic variables $z^r_{i,k}$ as follows. For each k and j, let $M_{k,j}$ be the set of minimal subsets of $\{(i,r) \mid i = 1,\ldots,n,\ r = 1,\ldots,\gamma_{i,k}\}$ with

$$\sum_{(i,r)\in I} \alpha_{i,j} z_{i,\{(k,r)\}} > \mu_{k,j}\ .$$

For each subset J of some set $I \in M_{k,j}$, we introduce a variable z_J modeling the monomial $\prod_{(i,r)\in J} z^r_{i,k}$. Let P^{****} denote the resulting polytope. Then P^{****} is isomorphic to a face of a boolean quadric polytope Q by [1] and P^{***} is a projection of P^{****}. Now the equation (13) implies $z_I = 0$ for all $I \in M_{k,j}$. The

set of constraints $z_I = 0$ for $I \in M_{k,j}$, for all k and j, induces a face F of P^{****}. In this face, the inequality

$$\sum_{i=1}^{n} \alpha_{i,j} \sum_{r=1}^{\gamma_{i,k}} z_{i,\{(k,r)\}} \leq \mu_{k,j}$$

is valid, so adding (13) induces a face F^* of F and hence of Q. This proves the first statement. The second statement follows from the fact that the cardinality of the set $M_{k,j}$ is polynomial for constant $\mu_{k,j}$. □

Example 4. Continuing Example 3, we have the linear constraints

$$2z^1_{1,1} + z^1_{2,1} + z^2_{2,1} + 2z^1_{3,1} = 2$$
$$2z^1_{1,2} + z^1_{2,2} + z^2_{2,2} + 2z^1_{3,2} = 2$$

and eliminate the minimal infeasible solutions by adding

$$z^1_{1,1}z^1_{2,1} = z^1_{1,1}z^2_{2,1} = z^1_{1,1}z^1_{3,1} = z^1_{2,1}z^1_{3,1} = z^2_{2,1}z^1_{3,1} = 0$$
$$z^1_{1,1}z^1_{2,1} = z^1_{1,1}z^2_{2,1} = z^1_{1,1}z^1_{3,1} = z^1_{2,1}z^1_{3,1} = z^2_{2,1}z^1_{3,1} = 0\ .$$

After linearizing all $13 + 10$ non-linear monomials of the problem and reducing it to a binary quadratic optimization problem according to [1, 2], the constraints

$$2z^1_{1,1} + z^1_{2,1} + z^2_{2,1} + 2z^1_{3,1} \geq 2$$
$$2z^1_{1,2} + z^1_{2,2} + z^2_{2,2} + 2z^1_{3,2} \geq 2$$

are face-inducing. □

References

1. Christoph Buchheim and Giovanni Rinaldi. Efficient reduction of polynomial zero-one optimization to the quadratic case. *SIAM Journal on Optimization*, 18(4):1398–1413, 2007.
2. Christoph Buchheim and Giovanni Rinaldi. Terse integer linear programs for boolean optimization. *Journal on Satisfiability, Boolean Modeling and Computation*, 6:121–139, 2009.
3. Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* A series of books in the mathematical sciences. W. H. Freeman and Company, New York, 1979.
4. Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Math.* Springer, Berlin, 1993.
5. Richard E. Rosenthal. *GAMS – A User's Guide*, 2007. Available at http://www.gams.com/docs/gams/GAMSUsersGuide.pdf.
6. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.
7. Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2, Ser. B):225–249, 2004.