Minimal conic quadratic reformulations and an optimization model

Ramez Kian [a,b,*], Emre Berk [c], Ülkü Gürler [d]

[a] Notingham Business School, Nottingham Trent University, Nottingham NG1 4FQ, UK
[b] Department of Industrial Engineering, University of Tabriz, Tabriz 51666, Iran
[c] Department of Management, Bilkent University, Ankara 06800, Turkey
[d] Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey

ARTICLE INFO

ABSTRACT

In this paper, we consider a particular form of inequalities which involves product of multiple variables with rational exponents. These inequalities can equivalently be represented by a number of conic quadratic forms called cone constraints. We propose an integer programming model and a heuristic algorithm to obtain the minimum number of cone constraints which equivalently represent the original inequality. The performance of the proposed algorithm and the computational effect of reformulations are numerically illustrated.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In this paper, we consider a particular form of inequalities of the *n-block-power-m* type as described in Definition 1. These are encountered in a wide range of optimization models in different fields such as engineering, finance, robust optimization and combinatorics. Special cases of them commonly appear in the constraints set of optimization models including convex power functions [1], antenna design, portfolio optimization, truss design and signal filter design (see [8]). In the following sections we will show that an *n-block-power-m* inequality can equivalently be expressed with a number of second-order (quadratic) cones (see Definition 2), which will be referred as *conic reformulation* in the rest of the paper. Reformulating the original optimization problem with inequalities of more general forms equivalently with cone constraints facilitates their solutions by user-developed programs or available commercial solvers. For example, when a linear optimization problem involves an inequality of the form given in (1), it may be difficult to solve. However, when the problem is expressed in a form that involves only quadratic cone constraints, then it belongs to the well known class of Second Order Cone Programming (SOCP), for which a vast number of solution approaches exist. Hence, the contribution of our work lies in an intermediate stage of an optimization problem where a general inequality given in the form above is converted to cone inequalities that are handled with more ease. We will also see

below that a conic reformulation results in creation of several simpler inequities that equivalently represent the original one, but such a reformulation is not unique. Hence from both practical and theoretical point of view it is desirable to obtain an equivalent representation with minimum number of conic constraints. We will refer to a conic reformulation with minimum possible number of conic constraints as the *minimal reformulation*. For the reformulation, we provide an iterative power reduction scheme which creates a new inequality at each iteration. We propose an Integer Programming (IP) model and a fast heuristic algorithm to obtain the minimum number of cone constraints which are equivalent to an *n-block power-m* inequality in its initial form. The performance of the proposed algorithm and the computational effect of reformulations are also illustrated.

The rest of the manuscript is organized as follows. In Section 2, we give preliminary definitions and a brief discussion of the related literature to demonstrate how an *n-block-power-m* inequality is convertible to cone constraints. In Section 3 we provide our reformulation scheme together with analytical results. Then we propose an IP model and a heuristic algorithm for reformulation. In Section 4 we present the performance of the proposed algorithm in generating minimal reformulations, and a short numerical study to illustrate the computational benefit of minimal reformulations. We conclude the paper with potential applications of our results in Section 5.

2. Preliminaries and literature review

In this section we provide the basic definitions needed for the reformulation of an *n-block-power-m* inequality. In line with the usual convention of the optimization models, throughout this

---

* Corresponding author at: Notingham Business School, Nottingham Trent University, Nottingham NG1 4FQ, UK.
 E-mail addresses: ramez.kian@ntu.ac.uk (R. Kian), eberk@bilkent.edu.tr (E. Berk), ulku@bilkent.edu.tr (Ü. Gürler).

article all variables are assumed to be non-negative. Also, $(.)^\top$ denotes the transpose of a vector $(.)$.

We begin by introducing the special form of inequalities mentioned above.

Definition 1. For $m, n \in \mathbb{N}$, an inequality of the form

$$t_0^{2^m} \le t_1^{r_1} t_2^{r_2} \cdots t_n^{r_n} \tag{1}$$

where $t_0 \in \mathbb{R}$, the variables $t_1, t_2, \ldots, t_n \in \mathbb{R}^+$ are non-identical and $r_1 + r_2 + \cdots + r_n = 2^m$ with $r_1, \ldots, r_n \in \mathbb{N}$, is an *n-block power-m* inequality.

Mathematically speaking, inequalities of the above form can appear directly or after an algebraic conversion of a variety of functions including logarithmic Tchebychev approximation, harmonic mean of a positive affine function, hypograph of the geometric mean, inequalities involving the sum of quadratic or linear fractions, and inequalities involving *p*-norms and rational powers (see [2,3]).

Definition 2. The second-order (Lorentz) cone, $\mathcal{C}^n$, $(n \ge 2)$ in $\mathbb{R}^n$ is defined as

$$\mathcal{C}^n = \left\{ (x_1, \ldots, x_n)^\top \in \mathbb{R}^n : x_1^2 \ge \sum_{i=2}^n x_i^2 \right\}. \tag{2}$$

The terms "quadratic cone" and "ice-cream cone" are other common terminologies for the definition above.

Definition 3. The rotated second-order cone, $\mathcal{C}_r^n$, $(n \ge 3)$, in $\mathbb{R}^n$ is defined as

$$\mathcal{C}_r^n = \left\{ (x_1, \ldots, x_n)^\top \in \mathbb{R}^n : x_1 x_2 \ge \sum_{i=3}^n x_i^2 \right\}. \tag{3}$$

Note that in SOCP models, a linear function is minimized over the intersection of an affine set and the Cartesian product of second-order cones (see [8] for further details).

Remark 1. It is easy to verify by simple algebra that if $\mathbf{x} = (x_1, x_2, x_3 \ldots, x_n)^\top \in \mathcal{C}_r^n$ for $(n \ge 3)$ then $\tilde{\mathbf{x}} = \left(\frac{x_1 + x_2}{2}, \frac{x_1 - x_2}{2}, x_3, \ldots, x_n \right)^\top \in \mathcal{C}^n$.

According to (3), if $n = 3$, a *2-block-power-1* inequality is the same as a 3-dimensional rotated cone, which will be referred to as a '*cone constraint*'. Formally, conic reformulation of an *n-block-power-m* inequality refers to equivalently expressing it by a number of distinct *2-block-power-1* inequalities.

Conic reformulation of a special case of an *n-block-power-m* inequality has been discussed in [2,4]. In particular they considered the case where $r_i = 1$, $i = 1, \ldots, n = 2^m$ and showed that the inequality

$$s_0^{2^k} \le s_1 s_2 \ldots s_{2^k}, \qquad k \ge 1, \tag{4}$$

where $s_0 \in \mathbb{R}$ and $s_1 \ge 0, \ldots, s_{2^k} \ge 0$, can be expressed equivalently by $2^k - 1$ cone constraints. This upper bound is based on the so-called "*tower-of-variables*" [4] construction. In this design, at each level ℓ, $\ell = 1, \ldots, k$, an auxiliary variable $w_{p,(\ell)}$ is associated to each pair of variables. Without loss of generality consider two consecutive variables $\{s_{2p-1}$ and $s_{2p}\}$ if $\ell = 1$, or $\{w_{2p-1,(\ell-1)}$ and $w_{2p,(\ell-1)}\}$ otherwise, $p = 1, \ldots, 2^{k-\ell}$. Then, the corresponding cone constraint $w_{p,(1)}^2 \le s_{2p-1} s_{2p}$ or $w_{p,(\ell)}^2 \le w_{2p-1,(\ell-1)} w_{2p,(\ell-1)}$ is added. Therefore, in the first level, $\ell = 1$, $2^{k-\ell}$ new inequalities are created. After replacement of the new auxiliary variables in lieu of their corresponding paired variables and a power reduction in both sides of (4), it is re-written as $s_0^{2^{k-1}} \le w_{1,(1)} w_{2,(1)} \ldots w_{2^{k-1},(1)}$. The latter inequality has the same

structure. Hence, the procedure continues in similar fashion in the following levels by creating $2^{k-\ell}$ new cone constraints at level ℓ resulting in the creation of $2^{k-1} + 2^k + \cdots + 1$ cone constraints in total.

The *tower-of-variables* construction is not unique due to the possible permutations in the order of the variables. Moreover, in a more general form as the *n-block-power-m* inequality, where some of the variables can be identical, the combinations of some variables are reducible to a linear inequality. For instance, $w^2 \le uu$ is a redundant cone constraint as it is equivalent to $w \le u$ for non-negative variables u and w.

In conic reformulation of an *n-block-power-m* inequality, we aim to obtain a minimal conic reformulation by excluding such redundancies and repetitions. To the best of our knowledge, this has not been addressed in the literature before. A result related to ours but restricted to a special case is given in [10] which establishes that $\rho \in \mathbb{N}$ number of cone constraints are needed to equivalently represent an inequality in form of $y_0^{2^\rho} \le y_1^{k_1} y_2^{k_2} y_3^{k_3}$ wherein $k_1, k_2, k_3 \in \mathbb{Z}^+$, $k_1 + k_2 + k_3 = 2^\rho$ and two of them are odd numbers.

3. Problem statement

Conic reformulation of an *n-block-power-m* inequality rests on a simple fact given in the following lemma.

Lemma 1. *The inequality* $t_0^{2^m} \le \prod_{i=1}^n t_i^{r_i}$ *is equivalent to,*

$$t_0^{2^m} \le t_{\iota_1}^{r_{\iota_1} - \alpha} t_{\iota_2}^{r_{\iota_2} - \alpha} w^{2\alpha} \prod_{i \ne \iota_1, \iota_2} t_i^{r_i}, \tag{5a}$$

$$w^2 \le t_{\iota_1} t_{\iota_2}, \tag{5b}$$

wherein $t_{\iota_1}^\alpha$ *and* $t_{\iota_2}^\alpha$ *are paired and* $0 \le \alpha \le \min(r_{\iota_1}, r_{\iota_2})$.

The following example illustrates how Lemma 1 is invoked successively to reformulate a *3-block-power-4* inequality.

Example 1. $t_0^8 \le t_1^2 t_2^3 t_3^3$:

$\Leftrightarrow t_0^8 \le t_2^1 t_3^3 w_1^4, \qquad\qquad w_1^2 \le t_1 t_2, \quad (\alpha_1 = 2);$

$\Leftrightarrow t_0^8 \le t_3^2 w_1^4 w_2^2, \qquad\qquad w_2^2 \le t_2 t_3, \quad (\alpha_2 = 1);$

$\Leftrightarrow t_0^8 \le w_1^4 w_3^4, \qquad\qquad\quad w_3^2 \le t_3 w_2, \quad (\alpha_3 = 2);$

$\Leftrightarrow t_0^2 \le w_1 w_3, \qquad\qquad\qquad\qquad\qquad (\alpha_4 = 4).$

The selection of α and the two variables which are paired together at each iteration are the key factors in constructing the set of cone constraints itself and its cardinality denoted by Λ. In the following, we provide an upper bound on the minimum number of iterations for the process above. To that end, first note that any positive integer number can be uniquely represented by the sum over powers of 2 as is used to obtain its binary representation. Let $\Omega(r)$ be the set of powers of 2 whose sum equals r, for $r \in \mathbb{N}$. For example, $\Omega(5) = \{1, 4\}$ and $\Omega(8) = \{8\}$ so $|\Omega(5)| = 2$ whilst $|\Omega(8)| = 1$.

Remark 2. The cardinality of $\Omega(r)$ equals the number of '1's in the binary representation of $r \in \mathbb{N}$.

A useful mathematical characterization of $\Omega(r)$ is provided below.

Remark 3. For $r \in \mathbb{N}$ assuming $|\Omega(0)| := 0$, we have $|\Omega(r)| = 1 + |\Omega(r - 2^{\lfloor \log_2 r \rfloor})|$.

Proposition 1. *An n-block power-m inequality can be represented by at most* $\Lambda = \sum_{i=1}^n |\Omega(r_i)| - 1$ *many 2-block-power-1 inequalities using* $\Lambda - 1$ *additional auxiliary variables.*

Proof. The proof rests on the iterative use of Lemma 1. Let $r_i^{(k)}$ be the updated value of ith exponent at iteration k, $i = 1, \ldots, n + k$, where any index $i > n$ corresponds to the new auxiliary variable at iteration $i - n$, which appeared as w in (5a) and (5b). By iteratively invoking Lemma 1, choosing an arbitrary pair of variables t_{i_1} and t_{i_2} and setting the value of α with a number in $\Omega(r_{i_1}^{(k)}) \cap \Omega(r_{i_2}^{(k)}) \neq \emptyset$ at iteration k, the updated exponents at any two successive iterations $k - 1$ and k differ only in three elements as $|\Omega(r_{i_1}^{(k)})| = |\Omega(r_{i_1}^{(k-1)})| - 1$, $|\Omega(r_{i_2}^{(k)})| = |\Omega(r_{i_2}^{(k-1)})| - 1$ and $|\Omega(r_{n+k}^{(k)})| = 1$. Thus, at each iteration the total cardinality of these sets is reduced by 1, which implies that the procedure will stop after $\Lambda = \sum_{i=1}^{n} |\Omega(r_i)| - 1$ iterations (i.e., when (5a) itself is a *2-block-power-1* inequality). □

3.1. Minimal conic reformulation: An IP model

As illustrated in Example 1, the successive application of Lemma 1 is terminated when the original *n-block-power-m* inequality reduces to two blocks of positive exponents, each being 2^{m-1}. In addition, from Proposition 1 we know that no more than Λ iterations are needed (using *powers-of −2* decomposition). Thus, we are able to propose an IP model to minimize the number of iterations where Lemma 1 is invoked in (6)–(15) with the following decision variables whose numbers are bounded by Λ.

Decision variables:
y_i^k: A binary variable which equals 1 if variable i is selected at iteration k; 0, otherwise.
R_i^k: The updated exponent of ith variable at the end of iteration k.
w^k: A binary variable which equals 0 if the initial *n-block-power-m* inequality has completely decomposed to *2-block-power-1* inequalities (cone constraints) before iteration k; 1, otherwise.
α_k: A non-negative integer which denotes the reduction amount in exponents of the selected variables (see Lemma 1).
z_i^k: An auxiliary non-negative integer variable for linearization purpose.

$$\min \Gamma = \sum_{k=1}^{\Lambda} w^k, \tag{6}$$

s.t.

$$R_i^0 = r_i \qquad i = 1, \ldots, n, \tag{7}$$

$$\alpha_0 = 0, \tag{8}$$

$$R_i^k = R_i^{k-1} - \alpha_k y_i^k, \qquad \begin{matrix} t = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+t, \end{matrix} \tag{9}$$

$$R_{n+k}^k = 2\alpha_k, \qquad k = 1, \ldots, \Lambda, \tag{10}$$

$$w^k \leq 2^{m-1} - \alpha_{k-1}, \qquad k = 1, \ldots, \Lambda, \tag{11}$$

$$2^{m-1}(1 - w^k) \leq \alpha_{k-1}, \quad k = 1, \ldots, \Lambda, \tag{12}$$

$$\sum_{i=1}^{n+k-1} y_i^k = 2w^k, \qquad k = 1, \ldots, \Lambda, \tag{13}$$

$$w^k, y_i^k \in \{0, 1\}, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k, \end{matrix} \tag{14}$$

$$\alpha_k, R_i^k \in \mathbb{Z}^+, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k. \end{matrix} \tag{15}$$

Constraint (9) is nonlinear due to the multiplication of α_k by y_i^k, which can be linearized as follows:

$$R_i^k = R_i^{k-1} - z_i^k, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k, \end{matrix} \tag{16}$$

$$z_i^k \leq 2^m y_i^k, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k, \end{matrix} \tag{17}$$

$$z_i^k \leq \alpha_k, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k, \end{matrix} \tag{18}$$

$$z_i^k \geq \alpha_k - (1 - y_i^k)2^m, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k, \end{matrix} \tag{19}$$

$$z_i^k \in \mathbb{Z}^+, \qquad \begin{matrix} k = 1, \ldots, \Lambda, \\ i = 1, \ldots, n+k. \end{matrix} \tag{20}$$

Constraints (7) and (8) initialize original values of the exponents and α, respectively. Constraint (9) calculates updated values of exponents while (10) assigns the exponent value of the newly defined auxiliary variables throughout the iterations. Constraints (11) and (12) are for determining if the stopping rule of the conic reformulation process is satisfied ($w^k = 0$) or not ($w^k = 1$) at iteration k. If not, constraint (13) forces the model to choose two additional variables to be paired. Finally, Constraints (14) and (15) define the type of our decision variables. The optimal objective value is bounded from above (i.e., $\Gamma^* \leq \Lambda$).

3.2. A heuristic reformulation

Although an exact solution can be found by the above IP model, to automate the user intervention, we propose a heuristic algorithm based on the iterative use of Lemma 1. We first define the *greatest-common-powers-of −2-cardinality* of two integer numbers as,

$$\mathcal{G}(k_1, k_2) := \left| \Omega(k_1) \cap \Omega(k_2) \right|. \tag{21}$$

A simple rule is employed in our heuristic at each iteration k for the selection of variables and the power by which they are reduced, α_k. That is, the pair of variables with maximum *greatest-common-powers-of −2-cardinality* of their exponents is selected; then α_k is set as the summation of the common *powers-of −2* elements within their exponents. This procedure is summarized in Algorithm 1. In the following example below, the impact of our algorithm in choosing variables and α_k is illustrated.

Algorithm 1 Conic reformulation of *n-block-power-m* inequality

Input: $m, n, (r_i, \ldots r_n)$ where $\sum_{i=1}^{n} r_i = 2^m$.
Output: Set of *2-block-power-1* inequalities.
1: $k \leftarrow 0, \quad \mathcal{S} \leftarrow \emptyset$ ▷ $\mathcal{S} =$ set of cone constraints
2: $R_i^0 \leftarrow r_i$ for $i \ldots n$.
3: **do**
4: $\quad (i_1, i_2) \leftarrow \arg \max\limits_{(i_1,i_2)|1 \leq i_1 < i_2 \leq n^{(t)}} \mathcal{G}(R_{i_1}^k, R_{i_2}^k)$
5: $\quad \mathcal{H} \leftarrow \Omega(R_{i_1}^k) \cap \Omega(R_{i_2}^k)$
6: $\quad k \leftarrow k + 1$
7: $\quad \alpha_k \leftarrow \frac{1}{2} \sum\limits_{h \in \mathcal{H}} h$
8: $\quad$ **if** $\alpha_k = 2^{m-1}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{t_0^2 \leq t_{i_1} t_{i_2}\}$ **else** $\mathcal{S} \leftarrow \mathcal{S} \cup \{t_{k+n}^2 \leq t_{i_1} t_{i_2}\}$
▷
$\quad t_{k+n}$ is a new auxiliary variable in addition to the existing ones in the initial *n-block-power-m*.
9: $\quad R_i^k \leftarrow R_i^{k-1} - \alpha_k$, for $i = i_1, i_2$
10: $\quad R_{n+k}^k \leftarrow 2\alpha_k$
11: **while** $(\alpha_k < 2^{m-1})$

Example 2. $t_0^8 \leq t_1^2 t_2^3 t_3^3$:

$\Leftrightarrow t_0^8 \leq t_1^2 w_1^6, \qquad\qquad w_1^2 \leq t_2 t_3 \quad (\alpha_1 = 3);$
$\Leftrightarrow t_0^8 \leq w_1^4 w_2^4, \qquad\qquad w_2^4 \leq t_1 w_1 \quad (\alpha_2 = 2);$
$\Leftrightarrow t_0^2 \leq w_1 w_2 \qquad\qquad\qquad\qquad (\alpha_3 = 4)$

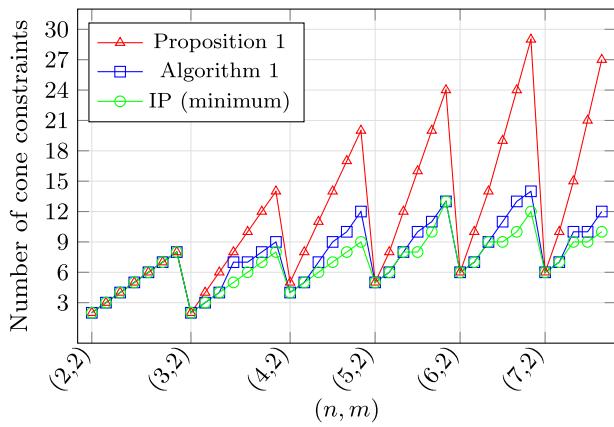which can be verified by our proposed IP model that is a minimal reformulation.

Fig. 1. Number of cone constraints obtained by Algorithm 1 vs. its upper bound/minimum.

Table 1
Computation time (in seconds) for the conic reformulated models.

Cone type:	Lorentz		Rotated	
Solver	Ex.1	Ex.2	Ex.1	Ex.2
CPLEX	18.96	21.82	22.40	29.36
Gurobi	2.89	5.00	15.97	21.30
Mosek	3.83	8.42	15.89	21.50
SDPT3	11.66	11.89	22.86	29.17
ECOS	2.52	2.98	15.49	20.89
SeDuMi	397.98	634.81	408.97	705.76

4. Numerical study

First we discuss the performance of Algorithm 1 in producing minimal reformulations. To this end, we used 35 instances of *n-block-power-m* inequalities ($2 \leq m, n \leq 7$), whose exponents were purposefully chosen to be *difficult*. (For the construction of the test bed, the reader is referred to Appendix.)

The number of cone constraints, required for the conic reformulation of the above-mentioned instances, are illustrated in Fig. 1.

According to our unreported additional tests, Algorithm 1 was able to obtain minimal reformulations for all instances obtained by minimization counterpart of (22)–(26). Its computation times are negligible in the order of milliseconds for the tested instances, while solving the IP model is prohibitively costly. For example, for our worst instance with $m = 7$ and $n = 7$, it took over 66 h for ILOG CPLEX 12.8 to stop even without obtaining the optimal solution (with relative MIPGap of 19.54%) and we could not optimally solve instances with $m > 8$ on a high performance Linux based server.

Apart from the way that *n-block-power-m* inequalities are minimally reformulated, in the next subsection we demonstrate the computational advantage of such a minimal formulation in different solver packages by an example.

4.1. A sample problem test

A valuable library of conic optimization problems has been gathered recently in (http://cblib.zib.de/). Among those, we have chosen the portfolio optimization problem with higher moment coherent risk measures (HMCR) as it is a convenient example for the issues discussed in this paper (see (32.a)–(32.g) in [7]). For that model we have set $p = 2.5$, $\alpha = 0.9$, $J = 1024$, $r_0 = 0.5\%$, $n = 50$ and r_{ij} is drawn from a lognormal distribution with parameters $\mu = 51.9$ and $\sigma = 103.1$ according to data description in [7]. Constraint $t \geq (w_1^p + \cdots + w_J^p)^{1/p}$ in that model can be re-written equivalently as $t \geq u_1 + \cdots + u_J$, $w_j^p \leq u_j t^{p-1}$ for $j = 1, \ldots, J$. The latter inequality is easily convertible to *3-block-power-*$\lceil \log_2 r \rceil$ where $p = r/s$ such that $r, s \in \mathbb{N}$. The *3-block-power-3* inequality formed from $r = 5$ and $s = 2$ here, was reformulated in Examples 1 and 2 differently with 3 and 4 cone constraints, respectively. The computations corresponding to these two formulations are denoted by Ex.1 and Ex.2 in Table 1.

In addition to the obtained rotated cone constraints, using (2), we have also modeled the problem with the Lorentz cones and have fed both into MATLAB via YALMIP interface (see [9]).

Each configuration was solved with three commercial solvers namely CPLEX, Gurobi, Mosek; and three free solvers namely, ECOS, SDPT3, SeDuMi (see [5,11,12], respectively) on a personal computer equipped with Intel Core 2 Duo Processor, 2.1 GHz and a RAM of 2 GB.

Each number shown in Table 1 is the average of the total computation time (including model creation and solving) over 6 replications of the same instance. Regardless of the solver employed, the model with a smaller set of cone constraints requires considerably less computation. Another crucial observation in this experiment is that the model formulated with second-order cone constraints computationally outperforms the one with rotated cone constraints. Therefore, it is better to further convert the obtained *2-block-power-1* inequalities to the Lorentz cone form rather than leaving this task to the solvers.

5. Conclusion

In this paper, we considered a special type of inequalities termed as *n-block-power-m* and proposed a reformulation algorithm to equivalently express it by conic inequalities. The performance of the reformulation is also demonstrated by numerical examples. Recently, a class of problems called *p-order conic optimization* is introduced by Mosek (although has not been put into practice in its interface yet) and it covers problems with constraints including *p*-norm. To the best of our knowledge (see [6]), the solver reduces these types of constraints to several second-order cones. We therefore expect that our proposed algorithm efficiently provides such reductions and may provide significant benefits if embedded in such packages.

Appendix

In order to generate the test bed with difficult instances, we have used an auxiliary IP model whose construction is based on Proposition 1 and Remark 2. Let $x_{ij} \in \{0, 1\}$, $i = 1, \ldots, n$, $j = 0, \ldots, m - 1$ be *j*th digit in the binary representation of r_i, $i = 1, \ldots, n$. That is, $r_i = (\overline{x_{i,m-1}, \ldots, x_{i,1}, x_{i,0}})_2$. Thus, according to Remark 2, Λ (defined in Proposition 1) can be written as (22) which is maximized subject to constraints (23)–(26) in the following IP model.

$$\max \Lambda = \left(\sum_{i=1}^{n} \sum_{j=0}^{m-1} x_{ij} \right) - 1, \tag{22}$$

s.t.

$$\sum_{j=0}^{m-1} x_{ij} \geq 1, \quad i = 1, \ldots, m, \tag{23}$$

$$\sum_{j=0}^{m-1} 2^j \left(\sum_{i=1}^{n} x_{ij} \right) = 2^m, \tag{24}$$

$$\sum_{i=1}^{n} x_{i0} \geq 2, \quad i = 1, \ldots, m, \tag{25}$$

$$x_{ij} \in \{0, 1\}. \tag{26}$$

Constraint (23) guarantees positiveness of the exponents and (24) assures that their summation equals 2^m to comply with the definition. Constraint (25) limits the model to obtain irreducible inequalities by having at least two odd exponents. By solving this IP for a given n, m and converting the obtained optimal binary solution x_{ij} to the decimal form, the desired *difficult* exponents instances are retrieved.

References

[1] M.S. Aktürk, A. Atamtürk, S. GüRel, A strong conic quadratic reformulation for machine-job assignment with controllable processing times, Oper. Res. Lett. 37 (3) (2009) 187–191.

[2] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (1) (2003) 3–51.

[3] A. Ben-Tal, A. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, Vol. 2, Siam, 2001.

[4] A. Ben-Tal, A. Nemirovski, On polyhedral approximations of the second-order cone, Math. Oper. Res. 26 (2) (2001) 193–205.

[5] A. Domahidi, E. Chu, S. Boyd, ECOS: An SOCP solver for embedded systems, in: European Control Conference, ECC, 2013, pp. 3071–3076.

[6] H.A. Friberg, Power cones in second-order cone form and dual recovery, 2017, https://docs.mosek.com/slides/2017/siopt/powcones.pdf (Accessed January 2019).

[7] P.A. Krokhmal, Higher moment coherent risk measures, Quant. Finance 7 (4) (2007) 373–387.

[8] M.S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra Appl. 284 (1–3) (1998) 193–228.

[9] J. Löfberg, Yalmip: A toolbox for modeling and optimization in matlab, in: Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[10] Y. Morenko, A. Vinel, Z. Yu, P. Krokhmal, On p-norm linear discrimination, European J. Oper. Res. 231 (3) (2013) 784–789.

[11] J. Sturm, Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones, Optim. Methods Softw. 11–12 (1999) 625–653, Version 105 available from http://fewcal.kub.nl/sturm.

[12] K.C. Toh, M. Todd, R.H. Tütüncü, Sdpt3 – A matlab software package for semidefinite programming, Optim. Methods Softw. 11 (1999) 545–581.