Second-Order Conic and Polyhedral Approximations of the Exponential Cone: Application to Mixed-Integer Exponential Conic Programs

Qing Ye · Weijun Xie

Abstract Exponents and logarithms are fundamental components in many important applications such as logistic regression, maximum likelihood, relative entropy, and so on. Since the exponential cone can be viewed as the epigraph of perspective of the natural exponential function or the hypograph of perspective of the natural logarithm function, many mixed-integer convex programs involving exponential or logarithm functions can be recast as mixed-integer exponential conic programs (MIECPs). However, unlike mixed-integer linear programs (MILPs) and mixed-integer second-order conic programs (MISOCPs), MIECPs are still under development. To harvest the past efforts on MILPs and MISOCPs, this paper presents second-order conic (SOC) and polyhedral approximation schemes for the exponential cone with application to MIECPs. To do so, we first extend and generalize existing SOC approximation approaches in the extended space, propose new scaling and shifting methods, prove approximation accuracies, and derive lower bounds of approximations. We then study the polyhedral outer approximation of the exponential cones in the original space using gradient inequalities, show its approximation accuracy, and derive a lower bound of the approximation. When implementing SOC approximations, we suggest learning the approximation pattern by testing smaller cases and then applying it to the large-scale ones; and for the polyhedral approximation, we suggest using the branch and cut method for MIECPs. Our numerical study shows that the proposed methods show speed-ups over solver MOSEK for MIECPs, and the scaling, shifting, and polyhedral outer approximation methods work very well.

1 Introduction

Conic programming has recently witnessed a rapid growth of interest in both industry and academia and has been successfully applied to many domains (see, e.g., [1, 20, 21, 29, 30]). Notably, the recent version of solver MOSEK [2] can solve quite large-scale continuous exponential conic programs (ECPs). However, mixed-integer exponential conic programs (MIECPs) are still under development. It is known that polyhedral approximation results and strong valid inequalities developed during the last one or two decades can significantly enhance the capability of existing solvers such as Gurobi and CPLEX for solving mixed-integer linear programs (MILPs) and mixed-integer second-order conic programs (MISOCPs). To harvest these past efforts on MILPs and MISOCPs, we study second-order conic (SOC) approximations and polyhedral approximations of the exponential cone in the hope of solving MIECPs efficiently.

Qing Ye
Virginia Tech, Blacksburg, VA
E-mail: yqing1@vt.edu

Weijun Xie
Virginia Tech, Blacksburg, VA
E-mail: wxie@vt.edu

1.1 Setting

This paper focuses on the following MIECP

$$\min_{\boldsymbol{x} \in X \subset \mathbb{Z}^t \times \mathbb{R}^{n-t}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} : \left(\boldsymbol{a}_{i1}^\top \boldsymbol{x} + b_{i1}, \boldsymbol{a}_{i2}^\top \boldsymbol{x} + b_{i2}, \boldsymbol{a}_{i3}^\top \boldsymbol{x} + b_{i3} \right) \in K_{\exp}(0), \forall i \in [m] \right\} \tag{1}$$

where set X is mixed-integer and compact, $t \in [0, n]$, and $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3 \in \mathbb{R}^{m \times n}, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3 \in \mathbb{R}^m$ are data. For a given $\alpha \in \mathbb{R}$, we let $K_{\exp}(\alpha)$ denote the parametric exponential cone

$$K_{\exp}(\alpha) = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{hyp} \log(\alpha) \right\} \tag{2a}$$

$$= \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_3/x_2, x_1/x_2) \in \text{epi} \exp(\alpha) \right\} \tag{2b}$$

where the interval $\mathcal{R} := [1/M, M]$ for some positive parameter $M > 1$, set $\text{hyp} \log(\alpha)$ denotes the hypograph of the logarithm function parameterized by α

$$\text{hyp} \log(\alpha) = \left\{ (\boldsymbol{x}, t) : \log(\boldsymbol{x}) \geq t - \alpha \right\},$$

and set $\text{epi} \exp(\alpha)$ denotes the epigraph of the exponential function parameterized by α

$$\text{epi} \exp(\alpha) = \left\{ (\boldsymbol{x}, t) : \exp(\boldsymbol{x} - \alpha) \leq t \right\}.$$

We remark that for the modeling purpose, one can assume that M can be ∞ and by default, we let $0 \times \infty = 0$ and $1/\infty = 0$, under which set $K_{\exp}(\alpha)$ remains to be closed; however, for the analytical purpose, our approximation results only hold when M is finite. Besides, the parameter $\alpha \in \mathbb{R}$ is meant to quantify the error of the proposed approximations. It is also worthy of mentioning that when $\alpha = 0$, i.e., no approximation error, the exponential cone $K_{\exp}(0)$ in (2) can be viewed as the hypograph of perspective of the logarithm function or the epigraph of perspective of the exponential function. These two interpretations motivate us to study distinct approximation schemes.

As the exponential cone $K_{\exp}(0)$ is fundamental to an MIECP, this paper plans to develop SOC approximations of the exponential cone $K_{\exp}(0)$ of the form

$$K_{sc}(n^s, m^s) = \left\{ \boldsymbol{x} \in \mathbb{R}^{n^s} : \left((\boldsymbol{a}_{i1}^s)^\top \boldsymbol{x} + b_{i1}^s, (\boldsymbol{a}_{i2}^s)^\top \boldsymbol{x} + b_{i2}^s, (\boldsymbol{a}_{i3}^s)^\top \boldsymbol{x} + b_{i3}^s \right) \in \mathcal{L}_3, \forall i \in [m^s] \right\}, \tag{3}$$

where the Lorentz cone is defined as $\mathcal{L}_{q+1} = \left\{ \boldsymbol{y} \in \mathbb{R}^{q+1} : y_{q+1} \geq \sqrt{\sum_{i \in [q]} y_i^2} \right\}$ with $q \in \mathbb{Z}_{++}$ and $\boldsymbol{A}_1^s, \boldsymbol{A}_2^s, \boldsymbol{A}_3^s \in \mathbb{R}^{m^s \times n^s}, \boldsymbol{b}_1^s, \boldsymbol{b}_2^s, \boldsymbol{b}_3^s \in \mathbb{R}^{m^s}$ are data. We would like to show that for any $\epsilon > 0$, there exists a second-order cone $K_{sc}(n^s, m^s)$ with $n^s + m^s = O(f^s(1/\epsilon, M))$ depending on $1/\epsilon$ and M such that

$$K_{\exp}(-\epsilon) \subseteq K_{sc}(n^s, m^s) \subseteq K_{\exp}(\epsilon),$$

where $f^s(1/\epsilon, M)$ is a polynomial function. According to the seminal work [6] on the polyhedral approximations of SOC programs, there exists a polyhedral approximation of the exponential cone $K_{\exp}(0)$ such that

$$K_p(n^p, m^p) = \left\{ \boldsymbol{x} \in \mathbb{R}^{n^p} : (\boldsymbol{a}_i^p)^\top \boldsymbol{x} \geq b_i^p, \forall i \in [m^p] \right\}, \tag{4}$$

where $\boldsymbol{A}^p \in \mathbb{R}^{m^p \times n^p}, \boldsymbol{b}^p \in \mathbb{R}^{m^p}$ are data. Using the results in [6], for any $\epsilon > 0$ polyhedral approximation such that

$$K_{\exp}(-\epsilon) \subseteq K_p(n^p, m^p) \subseteq K_{\exp}(\epsilon),$$

the size of polyhedral approximation should be in the order of the size of second-order cone $K_{sc}(n^s, m^s)$ multiplied by the $\log(1/\epsilon)$, i.e., $n^p + m^p = O((n^s + m^s) \log(1/\epsilon))$. Since this type of polyhedral approximations of SOC programs has been successfully implemented in solvers such as Gurobi and CPLEX, this paper mainly focuses on polyhedral approximation in the original space, i.e., we mainly consider $n^p = 3$ in (4).

1.2 Relevant Literature

Many interesting nonlinear programs can be recast as an ECP (see, e.g., [25] for an overview). Below are some examples:

– (Entropy) The hypograph of the entropy function $f(x) = -x\log(x)$ is exponential-conic representable

$$\operatorname{hyp} f := \{(x,t) : -x\log(x) \geq t\} = \{(x,t) : (1,x,t) \in K_{\exp}(0)\}.$$

– (Kullback-Leibler Divergence) The epigraph of the Kullback-Leibler divergence or relative entropy function $f(x,y) = x\log(x/y)$ is exponential-conic representable

$$\operatorname{epi} f := \{(x,y,t) : x\log(x/y) \leq t\} = \{(x,y,t) : (y,x,-t) \in K_{\exp}(0)\}.$$

– (Logistic Regression) Given n data points $\{(\boldsymbol{x}_i,y_i)\}_{i\in[n]} \subseteq \mathbb{R}^d \times \{0,1\}$, the logistic regression with L_1 penalty can be formulated as the following unconstrained optimization problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i\in[n]} \left[-y_i\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) - (1-y_i)\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}_i))\right] + \lambda\|\boldsymbol{\theta}\|_1,$$

where $h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = [1 + \exp(-\boldsymbol{\theta}^\top\boldsymbol{x}_i)]^{-1}$ denotes the sigmoid function. The logistic regression can be recast as the following ECP

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d,\boldsymbol{p}_1,\boldsymbol{p}_2} \left\{ \sum_{i\in[n]} t_i + \lambda\|\boldsymbol{\theta}\|_1 : \begin{array}{l} p_{i1} + p_{i2} = 1, \left(p_{i1}, 1, (1-2y_i)\boldsymbol{\theta}^\top\boldsymbol{x}_i - t_i\right) \in K_{\exp}(0), \forall i \in [n], \\ (p_{i2}, 1, -t_i) \in K_{\exp}(0), \forall i \in [n] \end{array} \right\}.$$

– (Geometric Programming) Geometric programming problems [8] are nonlinear optimization problems of the form

$$\min_{\boldsymbol{x}} \left\{ f_0(\boldsymbol{x}) : f_i(\boldsymbol{x}) \leq 1, \forall i \in [m], x_j \geq 0, \forall j \in [n] \right\}.$$

Above, function $f_i(\boldsymbol{x}) = \sum_{k\in[T_i]} c_{ik} \prod_{j\in[n]} x_j^{a_{ijk}}$ is a posynomial for each $i \in [0,m]$, where $T_i \in \mathbb{Z}_+$, $c_{ik} \in \mathbb{R}_+$, and $a_{ijk} \in \mathbb{R}$ for each $k \in [T_i]$ and $j \in [n]$. Letting $x_j = \exp(y_j)$ for all $j \in [n]$ and substituting back the expressions, the epigraph of the revised "posynomial" $f_i(\boldsymbol{y})$ is exponential-conic representable

$$\operatorname{epi} f_i = \left\{ (\boldsymbol{y},t) : \sum_{k\in[T_i]} p_{ik} \leq 1, \left(p_{ik}, 1, \boldsymbol{a}_{ik}^\top\boldsymbol{y} + \log(c_{ik}) - t\right) \in K_{\exp}(0), \forall k \in [T_i] \right\}.$$

Therefore, the geometric programming problems can be equivalently transformed as the following ECPs

$$\min_{\boldsymbol{y}} \left\{ t_0 : (\boldsymbol{y},t_0) \in \operatorname{epi} f_0, (\boldsymbol{y},1) \in \operatorname{epi} f_i, \forall i \in [n] \right\}.$$

The ECPs or MIECPs also exist in many areas such as finance, defense, and healthcare. For example, the works in manpower planning [18], electric vehicle charging management [10], Kullback-Leibler divergence constrained distributionally robust optimization [19] have been using ECP approaches. Many other works might not have been fully aware of but can be reformulated as ECPs or MIECPs. For instance, in [24], the authors used the limited resources on-hand to maximize the probability of searching for objects, where the hypograph of the searching probability, in fact, can be represented as an exponential cone. Thus, their problem can be formulated as an MIECP. In [27], the authors proposed an ECP for sparse resource allocation in control of spreading processes. Their proposed method could be applied to minimize the spreading rate of epidemics and wildfires. Recently, the work in [31] proposed a joint estimation and robustness optimization framework that could be modeled with ECPs. As an important class of ECP problems, geometric programming has a variety of applications, including telecommunication [11], circuit design [9], and power control [12].

Recently, several works focused on solving continuous ECPs. For example, solver SCS [22] is a first-order method using operator splitting and homogeneous self-dual embedding, which can handle symmetric cones as well as the power and exponential cones. In [25], the authors analyzed the theoretical properties of some algorithms, such as primal and dual barrier methods, and extended an interior-point conic solver ECOS [16] to solve continuous ECPs. Alfonso [23] is a conic solver for nonsymmetric cones, including power and exponential cones, where the implementation is based on a homogeneous interior-point method proposed by [26]. The authors in [15] proposed a new primal-dual interior-point algorithm for continuous ECPs with a novel higher-order search direction. The work [3] analyzed theoretical properties of the algorithm implemented in MOSEK and proposed a new one for solving nonsymmetric conic optimization such as ECPs. The literature of generic approaches for solving ECPs, especially MIECPS, is relatively sparse. It is known that MOSEK can solve MIECPs [15] as early as 2018, which is mainly based on the interior point method and branch and bound. Parallelly, the work in [13] presented Pajarito, a generic branch and bound algorithm with outer approximation using primal and dual information for solving mixed-integer convex problems involving positive semidefinite, second-order, and exponential cones. Different from their methods, our methods are based on second-order approximations and gradient-based polyhedral approximations.

Our works on SOC approximations are motivated but different from the two seminal ones. The first interesting work is [17], which studied the semidefinite approximations of the matrix logarithm. They proposed a special function, whose integral over a particular domain is equal to the logarithm function and is semidefinite programming representable. However, their analysis is quite limited to a particular function, and mixed-integer semidefinite programming is known to be notoriously difficult to solve. On the contrary, we propose a large family of generating functions, which are SOC representable, and whose integrals from -1 to 1 are equal to the logarithm function. Our analysis is much simplified, and given an approximate solution, we propose a new scaling method, which works the best in our numerical study. The second interesting work is [5], which studied the SOC approximations of the exponential function. We analyze their approximation errors and propose a new shifting method with a given approximate solution. The numerical study shows that the proposed shifting method works well while other exponential function-based methods can have numerical issues. Besides, different from both works, we also derive lower bounds for the approximation errors and study polyhedral outer approximation based on gradient inequalities.

1.3 Summary of Contributions

This paper generalizes the existing and develops new SOC approximations of the exponential cone (2). We also study polyhedral outer approximation based on gradient inequalities. Our main contributions are summarized as below:

(i) We propose a generic SOC approximation framework of the exponential cone based on the logarithmic form (2a). We prove the approximation accuracy using the Gaussian quadrature. We tailor the proof to three special classes of SOC approximation. In particular, our proposed scaling approximation scheme using an approximate solution is numerically demonstrated to work the best.

(ii) We prove the approximation accuracies of the SOC approximations of the exponential cone using the exponential form (2b). We show that the even-order Taylor expansions of the exponential function can be written as the sum of squares and disprove a SOC representation proposed by [5]. We develop a new shifting method using an approximate solution, which overcomes the numerical issues caused by other approximation schemes using the exponential form (2b).

(iii) We study the minimum number of variables and SOC constraints needed to approximate the exponential cone to a desirable accuracy, i.e., study a lower bound of SOC approximations for the exponential cone.

(iv) We propose a polyhedral outer approximation of the exponential cone in the original space using gradient inequalities and study its upper and lower bounds for the number of inequalities to achieve a desirable approximation accuracy.

(v) Our numerical study shows that the proposed scaling, shifting, and polyhedral outer approximation methods outperform solver MOSEK for solving MIECPs and can achieve up to 20 times speed-ups.

The main theoretical approximation complexities of this paper are displayed in Table 1.

Table 1: Summary of Approximation Methods of the Exponential Cone

	SOC Approximations[i]				Polyhedral Approximation[i]
Methods	Logarithm-based [iii]				Original Space
	Example 1	Example 2	Example 3		
Complexity[ii]	$O(\sqrt{M}$ $+ \log(\sqrt{M^3/\epsilon}))$	$O(\log\log(M)$ $+\sqrt{\log(M) + \log(1/\epsilon)})$	$O(\log(1-\delta)/\log(\delta)$ $+ \log(\epsilon)/\log(\delta))$		$\Theta(\log(M) + 1/\sqrt{\epsilon})$
Outer Approximation	No	No	No		Yes
	Exponential-based [iv]				
Methods	Section 3.1	Section 3.2	Section 3.1 Shift	Section 3.2 Shift	
Complexity[ii]	$O(\log(M^2 \log M)$ $+ \log(1/\epsilon))$	$O(\log(M^2 \log(M)/s)$ $+ \log(1/\epsilon)/s)$	$O(\log(\delta^2/\epsilon))$	$O(\log(\delta/s)$ $+ \log(1/\epsilon)/s)$	
Outer Approximation	Yes	No	Yes	No	
Lower Bound	$\Omega(1 + \log\log(M)/\log(1/\epsilon))$				$\Omega(\log\log(M) + \log(1/\epsilon))$

[i] ϵ is the approximation accuracy; [ii] Complexity is the number of variables, SOC and linear constraints needed;
[iii] $\delta \in (0,1)$ is quality of the approximate solution $\widehat{x}_1/\widehat{x}_2$ such that $1 - \delta \leq (x_1/x_2)/(\widehat{x}_1/\widehat{x}_2) \leq 1 + \delta$;
[iv] $\delta > 0$ is quality of the approximate solution $\widehat{x}_3/\widehat{x}_2$ such that $|x_3/x_2 - \widehat{x}_3/\widehat{x}_2| \leq \delta$.

Notation. The following notation is used throughout the paper. We use bold letters (e.g., $\boldsymbol{x}, \boldsymbol{A}$) to denote vectors and matrices and use corresponding non-bold letters to denote their components. Given an integer n, we let $[n] := \{1, 2, \ldots, n\}$, $[0, n] := \{0, 1, 2, \ldots, n\}$, and use $\mathbb{R}^n_+ := \{\boldsymbol{x} \in \mathbb{R}^n : x_i \geq 0, \forall i \in [n]\}$. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, its epigraph and hypograph are $\operatorname{epi} f = \{(\boldsymbol{x}, t) : f(\boldsymbol{x}) \leq t\}$ and $\operatorname{hyp} f = \{(\boldsymbol{x}, t) : f(\boldsymbol{x}) \geq t\}$, respectively. For the sake of simplicity, the logarithm function and exponential function, if not being specified, are both meant to be natural ones in this paper. Additional notation will be introduced as needed.

Organization. The remainder of the paper is organized as follows. Section 2 develops and analyzes the approximation schemes using the logarithmic form (2a). Section 3 develops and analyzes the approximation schemes using the exponential form (2b). Section 4 shows lower bounds of approximations. Section 5 shows polyhedral outer approximation using gradient inequalities and proves the upper and lower bounds of approximations. Section 6 shows the numerical illustration. Finally, Section 7 concludes this paper.

2 Approximations Using the Logarithmic Form (2a)

In this section, we study approximations of the exponential cone using the logarithm form (2a). The key idea is to find a family of functions whose integrals are equal to the logarithm function.

2.1 Approximating the Logarithm Function Based on Integrals of Generating Functions

Since it is unlikely that the logarithm function $\log(x)$ is SOC representable, we plan to approximate the logarithm $\log(x)$ with $x \in \mathcal{R}$ by rewriting it as the integral of an SOC representable function.

To begin with, let us define the generating functions for the logarithm function.

Definition 1 (Generating Functions) A function $\phi : [-1, 1] \times \mathcal{R} \to \mathbb{R}$ is a "*generating function*" for the logarithm function $\log(x)$ if it satisfies

(a) For any given $t \in [-1, 1]$, function $\phi(t, x)$ is concave and continuous, and its hypograph

$$\operatorname{hyp} \phi(t) := \{(x, v) : \phi(t, x) \geq v\}$$

is SOC representable in the domain $\mathcal{R}$; and

(b) For any $x \in \mathcal{R}$, the following identity must hold

$$\int_{-1}^{1} \phi(t, x)dt = \log(x) + \text{const.} \tag{5}$$

Note that the generating functions are useful to derive SOC approximations of the logarithm function since the one-dimensional integration (5) admits efficient approximations (e.g., using Gaussian quadratures). In fact, there exist many such generating functions. In Section 2.1.1-Section 2.1.3, we show three examples, which are quite effective in our numerical study.

Although generating functions in Definition 1 are SOC representable, it still remains to address the left-hand integration of (5). Fortunately, the modern numerical methods are quite mature in one-dimensional integration. Thus, we propose to approximate the left-hand integration of (5) using well-known Gaussian quadruple (see, e.g., Chapter 19 [28]). Suppose $\{t_k\}_{k \in [N]} \subseteq [-1, 1]$ be N quadrature points and their corresponding positive weights $\{w_k\}_{k \in [N]}$. Then the logarithm function $\log(x)$ can be tightly approximated by

$$\int_{-1}^{1} \phi(t, x)dt = \log(x) + \text{const} \approx \sum_{k \in [N]} w_k \phi(t_k, x).$$

Accordingly, the hypograph of $\log(x)$ can be approximated by

$$\text{hyp} \log(\alpha)|_{\alpha=0} := \{(x, \nu) \in \mathcal{R} \times \mathbb{R} : \log(x) \geq \nu - \alpha|_{\alpha=0}\}$$

$$\approx \left\{ (x, \nu) \in \mathcal{R} \times \mathbb{R} : \sum_{k \in [N]} w_k v_k \geq \nu + \text{const}, (x, v_k) \in \text{hyp} \, \phi(t_k), \forall k \in [N] \right\} := \widehat{H}_{\phi, N}, \tag{6}$$

where constant α is useful to characterize the approximation accuracy, and the hypograph $\text{hyp} \, \phi(t)$ is supposed to be SOC representable. The Gaussian quadrature is quite accurate to approximate one-dimensional integration. In fact, according to Chapter 19 [28], the error of Gaussian quadrature decays geometrically.

Theorem 1 *Suppose that there exists some $\rho > 1$ such that*

$$\sup_{x \in \mathcal{R}, z \in \mathbb{C}} \left\{ |\phi(z, x)| : z = \frac{\rho}{2} \exp(i\theta) + \frac{\rho^{-1}}{2} \exp(-i\theta) : \theta \in [0, 2\pi] \right\} \leq L < \infty.$$

Then the approximation accuracy of Gaussian quadrature satisfies

$$\left| \log(x) + const - \sum_{k \in [N]} w_k \phi(t_k, x) \right| \leq \frac{64 L \rho^{-2N}}{15(\rho^2 - 1)}.$$

Thus, achieving the ϵ-approximation accuracy with $\epsilon > 0$, we can choose the number of points $N = O(\log_\rho(L/\epsilon))$ such that $\text{hyp} \log(-\epsilon) \subseteq \widehat{H}_{\phi, N} \subseteq \text{hyp} \log(\epsilon)$.

The key ingredients in Theorem 1 are two parameters ρ and L, which require effort to derive. Fortunately, we can derive these parameters for the examples in Section 2.1.1-Section 2.1.3 explicitly.

2.1.1 Example 1

Example 1 Suppose $\phi_1(t, x) = \frac{a(t+1)^{a-1}(x-1)}{2^a + (t+1)^a(x-1)}$ for some positive parameter $a > 0$. Then we have $\int_{-1}^{1} \phi_1(t, x) dt = \log(x)$ and for any $t \in [-1, 1]$, the hypograph of $\phi_1(t, \cdot)$ is SOC representable, i.e.,

$$\text{hyp } \phi_1(t) := \left\{ (x, v) \in \mathcal{R} \times \mathbb{R} : \begin{array}{c} r_1 = a - (t+1)v \geq 0, \\ r_2 = a(t+1)^{a-1}(x-1) - 2^a v \geq 0, \\ (v\sqrt{2^{a+2}(t+1)}, r_1 - r_2, r_1 + r_2) \in \mathcal{L}_3 \end{array} \right\}. \tag{7}$$

Next, we study the number of variables and SOC constraints needed to achieve the ϵ-approximation accuracy using Example 1.

Corollary 1 *For the generating function in Example 1 with $a = 1$, we can choose $\rho = (M+1)/(M-1) + 2/\sqrt{M}$, and*

$$L = \frac{1}{\frac{M+1}{M-1} - \frac{\rho}{2} - \frac{\rho^{-1}}{2}}.$$

Thus, achieving the ϵ-approximation accuracy requires the number of points $N = O(\sqrt{M}\log(\sqrt{M^3/\epsilon}))$ such that $\text{hyp}\log(-\epsilon) \subseteq \widehat{H}_{\phi_1,N} \subseteq \text{hyp}\log(\epsilon)$. That is, there should be $O(\sqrt{M}\log(\sqrt{M^3/\epsilon}))$ number of variables and SOC constraints in the representation of set $\widehat{H}_{\phi_1,N}$, which admits

$$\widehat{H}_{\phi_1,N} = \left\{ (x, \nu) \in \mathcal{R} \times \mathbb{R} : \begin{array}{c} \sum_{k \in [N]} w_k v_k \geq \nu, \\ r_{1k} = a - (t_k + 1)v_k \geq 0, \forall k \in [N], \\ r_{2k} = a(t_k + 1)^{a-1}(x-1) - 2^a v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{2^{a+2}(t_k+1)}, r_{1k} - r_{2k}, r_{1k} + r_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{array} \right\}. \tag{8}$$

Proof: We observe that $|z|$ can be upper bounded by

$$|z| = \left[\frac{1}{4}\left(\rho + \rho^{-1}\right)^2 \cos^2\theta + \frac{1}{4}\left(\rho - \rho^{-1}\right)^2 \sin^2\theta \right]^{1/2} \leq \frac{\rho}{2} + \frac{\rho^{-1}}{2}.$$

Then, we have

$$|\phi(z, x)| = \left| \frac{x-1}{2 + (z+1)(x-1)} \right| = \frac{1}{\left|\frac{x+1}{x-1} + z\right|} \leq \frac{1}{\left|\left|\frac{x+1}{x-1}\right| - |z|\right|} \leq \frac{1}{\frac{M+1}{M-1} - \frac{\rho}{2} - \frac{\rho^{-1}}{2}} := L,$$

where the first inequality is because of triangle inequality and the second one holds since $x \in \mathcal{R}$ and we choose

$$\rho = (M+1)/(M-1) + 2/\sqrt{M} \in \left(1, \frac{M+1}{M-1} + \sqrt{\left(\frac{M+1}{M-1}\right)^2 - 1}\right)$$

such that $|x+1|/|x-1| \geq (M+1)/(M-1) > |z|$. According to the choice of ρ and L, we have

$$\frac{L}{\rho^2 - 1} = \frac{1}{(\rho^2 - 1)(\frac{\rho^2 - 1}{2\rho} - \frac{2}{\sqrt{M}})} = \frac{2\rho}{(\rho^2 - 1)[(\rho - \frac{2}{\sqrt{M}})^2 - \frac{4}{M} - 1]} = \frac{\rho M(M-1)^2}{(\rho^2 - 1)(4M - 2)} \leq \frac{\rho(M-1)^3}{16} \leq \frac{3M^3}{16}.$$

We then split the proof into the following two cases:

Case 1. If $M \in [1, 4]$, then $\rho \geq 7/3$ and $L \leq 7/2$. The result holds.

Case 2. If $M \geq 4$, we have

$$\frac{\log\left(\sqrt{64L/15(\rho^2-1)\epsilon}\right)}{\log(\rho)} \leq \frac{\log\left(\sqrt{4M^3/5\epsilon}\right)}{\log\left(1+2/\sqrt{M}\right)} \leq \sqrt{M}\log\left(\sqrt{4M^3/5\epsilon}\right),$$

where the first inequality is due to $L/(\rho^2-1) \leq 3M^3/16$ and $t/(1+t) \leq \log(1+t)$ for any $t > 0$ and the second one is due to $M \geq 4$. Therefore, we can choose $N = O(\sqrt{M}\log(\sqrt{M^3/\epsilon}))$.

Finally, the SOC representation of $\widehat{H}_{\phi_1,N}$ follows from the fact that the hypograph of $\phi_1(t_k, x)$ is SOC representable for each $k \in [N]$. □

The result in Corollary 1 holds only when $a = 1$ in Example 1. Our numerical study shows that letting $a = 1$ performs nearly the best among all the testing instances. For general $a > 0$, we are unable to derive closed-form ρ and L and thus leave it to the interested readers.

2.1.2 Example 2

Example 2 Suppose $\phi_2(t, x) = \frac{2^s(x^{1/2^s}-1)}{2+(t+1)(x^{1/2^s}-1)}$ for some positive parameter $a > 0$ and positive integer $s \in \mathbb{Z}_{++}$. Then we have $\int_0^1 \phi_2(t, x)dt = \log(x)$ and for any $t \in [0, 1]$, the hypograph of $\phi_2(t, \cdot)$ is SOC representable, i.e.,

$$\text{hyp }\phi_2(t) := \left\{ (x, v) \in \mathcal{R} \times \mathbb{R} : \begin{array}{l} r_i \geq 0, \forall i \in [s], \\ (2r_1, x-1, x+1) \in \mathcal{L}_3, \\ (2r_{i+1}, r_i - 1, r_i + 1) \in \mathcal{L}_3, \forall i \in [s-1], \\ \gamma_1 = 2^s - (t+1)v \geq 0, \\ \gamma_2 = r_s - 1 - 2^{1-s}v \geq 0, \\ (v\sqrt{2^{3-s}(t+1)}, \gamma_1 - \gamma_2, \gamma_1 + \gamma_2) \in \mathcal{L}_3 \end{array} \right\}. \tag{9}$$

This generating function was studied in [17] and was shown to be semidefinite conic representable.

Next, we study the number of variables and SOC constraints needed to achieve the $\epsilon-$approximation accuracy using Example 2.

Corollary 2 *For the generating function in Example 2, we can choose $\rho = (M^{1/2^s}+1)/(M^{1/2^s}-1)$, $s = N$, and*

$$L = \frac{2^s}{\frac{M^{1/2^s}+1}{M^{1/2^s}-1} - \frac{\rho}{2} - \frac{\rho^{-1}}{2}}.$$

Thus, achieving the $\epsilon-$approximation accuracy requires the number of points $N = O(\log\log(M) + \sqrt{\log(M) + \log(1/\epsilon)})$. That is, there should be $O(\log\log(M) + \sqrt{\log(M) + \log(1/\epsilon)})$ number of variables and SOC constraints in the representation of set $\widehat{H}_{\phi_2,N}$, which admits

$$\widehat{H}_{\phi_2,N} = \left\{ (x, \nu) \in \mathcal{R} \times \mathbb{R} : \begin{array}{l} \sum_{k \in [N]} w_k v_k \geq \nu, r_i \geq 0, \forall i \in [s], \\ (2r_1, x-1, x+1) \in \mathcal{L}_3, \\ (2r_{i+1}, r_i - 1, r_i + 1) \in \mathcal{L}_3, \forall i \in [s-1], \\ \gamma_{1k} = 2^s - (t_k+1)v_k \geq 0, \forall k \in [N], \\ \gamma_{2k} = r_s - 1 - 2^{1-s}v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{2^{3-s}(t_k+1)}, \gamma_{1k} - \gamma_{2k}, \gamma_{1k} + \gamma_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{array} \right\}. \tag{10}$$

Proof: We choose L in the following way

$$|\phi(z,x)| = \left| \frac{2^s(x^{1/2^s}-1)}{2+(z+1)(x^{1/2^s}-1)} \right| = \frac{2^s}{\left| \frac{x^{1/2^s}+1}{x^{1/2^s}-1} + z \right|} \leq \frac{2^s}{\left| \left| \frac{x^{1/2^s}+1}{x^{1/2^s}-1} \right| - |z| \right|} \leq \frac{2^s}{\frac{M^{1/2^s}+1}{M^{1/2^s}-1} - \frac{\rho}{2} - \frac{\rho^{-1}}{2}} := L,$$

where the first inequality is because of triangle inequality and the second one holds by choosing

$$\rho = (M^{1/2^s}+1)/(M^{1/2^s}-1) \in \left(1, \frac{M^{1/2^s}+1}{M^{1/2^s}-1} + \sqrt{\left(\frac{M^{1/2^s}+1}{M^{1/2^s}-1} \right)^2 - 1} \right)$$

such that

$$\frac{|x^{1/2^s}+1|}{|x^{1/2^s}-1|} \geq \frac{M^{1/2^s}+1}{M^{1/2^s}-1} > |z|.$$

Then we have

$$\frac{L}{\rho^2-1} = \frac{2^{s+1}\rho}{(\rho^2-1)^2} = \frac{2^{s-3}(M^{1/2^s}+1)(M^{1/2^s}-1)^3}{M^{2/2^s}} \leq 2^{s-3}M^{2/2^s}.$$

Letting $s = N$, we arrive at

$$\frac{64L\rho^{-2N}}{15(\rho^2-1)} \leq \frac{2^{N+3}M^{2/2^N}\rho^{-2N}}{15} \leq \frac{2^{-2N^2-N+3}M\log^{2N}(M)}{15} \leq \epsilon,$$

where the second inequality is due to

$$\frac{M^{1/2^N}-1}{M^{1/2^N}+1} = \tanh\left(\frac{1}{2}\log\left(M^{1/2^N} \right) \right) \leq \frac{\log(M)}{2^{N+1}}.$$

Solving the following inequality

$$2\log(2)N^2 + (\log(2) - 2\log\log(M))N + \log(15\epsilon/8M) \geq 0,$$

we have

$$N \geq \left[2\log\log(M) - \log(2) + \sqrt{[\log(2) - 2\log\log(M)]^2 - 8\log(2)\log(15\epsilon/8M)} \right] /4\log(2).$$

Thus, $N = s = O(\log\log(M) + \sqrt{\log(M) + \log(1/\epsilon)})$.

Finally, the SOC representation of $\widehat{H}_{\phi_2,N}$ follows from the fact that the hypograph of $\phi_2(t_k,x)$ is SOC representable for each $k \in [N]$. □

The generating function $\phi_2(t,x)$ has also been studied in [17]. We differentiate from their work in the following three aspects: (i) We prove Corollary 2 from a different angle and significantly simplify the proof; (ii) We focus on the SOC approximation, while [17] focus on semidefinite conic approximation; and (iii) The proof technique here can be applied to other generating functions.

2.1.3 Example 3

Example 3 Suppose $\phi_3(t, x) = \frac{(x/\widehat{x}-1)}{2+(t+1)(x/\widehat{x}-1)}$ for $\widehat{x} \in \mathcal{R}$. Then we have $\int_{-1}^{1} \phi_3(t, x)dt = \log(x) - \log(\widehat{x})$ and for any $t \in [-1, 1]$, the hypograph of $\phi_3(t, \cdot)$ is SOC representable, i.e.,

$$\text{hyp}\,\phi_3(t) := \left\{ (x, v) \in \mathcal{R} \times \mathbb{R} : \begin{matrix} r_1 = 1 - (t+1)v \geq 0, \\ r_2 = (x/\widehat{x} - 1) - 2v \geq 0, \\ (v\sqrt{8(t+1)}, r_1 - r_2, r_1 + r_2) \in \mathcal{L}_3 \end{matrix} \right\}. \tag{11}$$

In this example, if we obtain a near-optimal solution $\widehat{x}$, we can scale the variable x by $\widehat{x}$. The scaling method enables us to focus on the neighborhood around the solution $\widehat{x}$ and meanwhile maintains the SOC representability of the hypograph of the scaled generating function.

Next, we study the number of variables and SOC constraints needed to achieve the ϵ−approximation accuracy using Example 3.

Corollary 3 *For the generating function in Example 3, suppose $\widehat{x}$ is a near-optimal solution such that $1-\delta \leq x/\widehat{x} \leq 1 + \delta$ with $\delta \in (0, 1)$ and we choose $\rho = (2 - \delta)/\delta$ and $L = ((2 - \delta)/\delta - \rho/2 - \rho^{-1}/2)^{-1}$. Then, achieving the ϵ−approximation accuracy requires the number of points $N = O(\log(1 - \delta)/\log(\delta) + \log(\epsilon)/\log(\delta))$ such that $\text{hyp}\log(-\epsilon) \subseteq \widehat{H}_{\phi_3,N} \subseteq \text{hyp}\log(\epsilon)$. That is, there should be $O(\log(1 - \delta)/\log(\delta) + \log(\epsilon)/\log(\delta))$ number of variables and SOC constraints in the representation of set $\widehat{H}_{\phi_3,N}$, which admits*

$$\widehat{H}_{\phi_3,N} = \left\{ (x, \nu) \in \mathcal{R} \times \mathbb{R} : \begin{matrix} \sum_{k \in [N]} w_k v_k \geq \nu - \log(\widehat{x}), \\ r_{1k} = 1 - (t_k + 1)v_k \geq 0, \forall k \in [N], \\ r_{2k} = (x/\widehat{x} - 1) - 2v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{8(t_k+1)}, r_{1k} - r_{2k}, r_{1k} + r_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{matrix} \right\}. \tag{12}$$

Proof: We derive L for the generating function in Example 3 as follows

$$|\phi(z, x)| = \left| \frac{(x/\widehat{x} - 1)}{2 + (z + 1)(x/\widehat{x} - 1)} \right| = \frac{1}{\left| \frac{x/\widehat{x}+1}{x/\widehat{x}-1} + z \right|} \leq \frac{1}{\left| \left| \frac{x/\widehat{x}+1}{x/\widehat{x}-1} \right| - |z| \right|} \leq \frac{1}{\frac{2-\delta}{\delta} - \frac{\rho}{2} - \frac{\rho^{-1}}{2}} = L,$$

where the first inequality is because of triangle inequality and the second one holds by choosing

$$\rho = (2 - \delta)/\delta \in \left(1, \frac{2 - \delta}{\delta} + \sqrt{\left(\frac{2 - \delta}{\delta}\right)^2 - 1} \right)$$

such that

$$\frac{|x/\widehat{x} + 1|}{|x/\widehat{x} - 1|} \geq \frac{2 - \delta}{\delta} > |z|.$$

Then, we get

$$\frac{L}{\rho^2 - 1} = \frac{2\rho}{(\rho^2 - 1)^2} = \frac{\delta^3(2 - \delta)}{8(1 - \delta)^2}.$$

Now let

$$\frac{64L\rho^{-2N}}{15(\rho^2 - 1)} = \frac{8\delta^{2N+3}}{15(2 - \delta)^{2N-1}(1 - \delta)^2} \leq \epsilon,$$

and we have

$$N \geq \frac{\log(\delta^3(2 - \delta)) - \log(15(1 - \delta)^2\epsilon/8)}{2\log((2 - \delta)/\delta)}.$$

Therefore, we can choose $N = O(\log(1 - \delta)/\log(\delta) + \log(\epsilon)/\log(\delta))$.

The SOC representation of $\widehat{H}_{\phi_3,N}$ follows from the fact that the hypograph of $\phi_3(t_k, x)$ is SOC representable for each $k \in [N]$. □

For Corollary 3, we remark that (i) if δ is close to 1 (i.e., the solution quality of $\widehat{x}$ is quite low), then $N = O(\log(1 - \delta)/\log(\delta))$, implying that the analysis may not be tight; (ii) if $\delta \to \epsilon$ (i.e., the solution quality is good), then $N = O(\log(\epsilon)/\log(\delta))$; particularly, if $\delta = \epsilon^{1/t}$ for some positive $t \geq 1$, then we only need $N = O(t)$ points to achieve the ϵ-approximation accuracy. This result, together with Figure 1 may explain why Example 3 works so well in our numerical study.

2.1.4 Comparisons of Example 1-Example 3

The numerical comparison of Example 1-Example 3 versus second-order Taylor approximation to approximate the logarithm function can be found in Figure 1, where we let $\widehat{x} = 2$ for Example 3. All three examples have better approximation performance than the second-order Taylor approximation. By applying the scaling method, Example 3 outperforms the other methods, which implies even two-point Gaussian quadrature can be a good approximation of $\log(x)$ as $\log(x) \approx \log(\widehat{x}) + \phi_3(\sqrt{1/3}, x) + \phi_3(-\sqrt{1/3}, x)$. In practice, we can run heuristics to obtain a good-quality $\widehat{x}$.
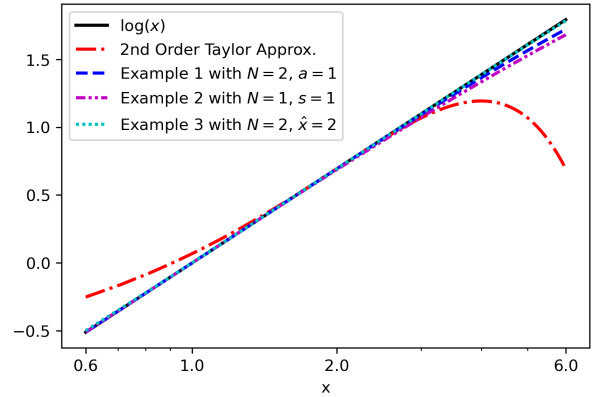

Fig. 1: A Comparison of the Second-order Taylor Approximation and Example 1-Example 3 using Two SOC Constraints to Approximate the Logarithm Function in the Domain $[0.6, 6.0]$

2.2 Approximating the Exponential Cone

Since the exponential cone in the form of (2a) can be viewed as the hypograph of perspective of the logarithm, the SOC approximations in the previous subsection can be directly applied to the exponential cone (2a). This result is summarized below:

Theorem 2 *Suppose that there exists some $\rho > 1$ such that*

$$\sup_{x \in [1/M^2, M^2], z \in \mathbb{C}} \left\{ |\phi(z, x)| : z = \frac{\rho}{2} \exp(i\theta) + \frac{\rho^{-1}}{2} \exp(-i\theta) : \theta \in [0, 2\pi] \right\} \leq L < \infty.$$

Then if the number of points $N = O(\log_\rho(L/\epsilon))$ and set

$$\widehat{K}^{sc}_{\phi,N} := \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \sum_{k \in [N]} w_k v_k \geq x_3 + const \cdot x_2, (x_1/x_2, v_k/x_2) \in \text{hyp } \phi(t_k), \forall k \in [N] \right\},$$

the following approximation result holds: $K_{\exp}(-\epsilon) \subseteq \widehat{K}^{sc}_{\phi,N} \subseteq K_{\exp}(\epsilon)$.

Proof: According to (6), for any α, we have

$$K_{\exp}(\alpha) = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{hyp } \log(\alpha) \right\}$$

and

$$\widehat{K}^{sc}_{\phi,N} = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \widehat{H}_{\phi,N} \right\}.$$

Thus, according to Theorem 1, for any $\epsilon > 0$, letting $N = O(\log_\rho(L/\epsilon))$, we have $K_{\exp}(-\epsilon) \subseteq \widehat{K}^{sc}_{\phi,N} \subseteq K_{\exp}(\epsilon)$ being equivalent to $\text{hyp } \log(-\epsilon) \subseteq \widehat{H}_{\phi,N} \subseteq \text{hyp } \log(\epsilon)$.

For the SOC approximation, we recall that

$$\widehat{H}_{\phi,N} = \left\{ (x,\nu) \in \mathcal{R} \times \mathbb{R} : \sum_{k \in [N]} w_k v_k \geq \nu + \text{const}, (x,v_k) \in \text{hyp}\,\phi(t_k), \forall k \in [N] \right\}.$$

Hence, we have

$$\widehat{K}_{\phi,N}^{sc} = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \sum_{k \in [N]} w_k v_k \geq x_3/x_2 + \text{const}, (x_1/x_2, v_k) \in \text{hyp}\,\phi(t_k), \forall k \in [N] \right\}.$$

Letting $v_k := v_k x_2$, we arrive at the desired formulation. Finally, since set $\text{hyp}\,\phi(t_k)$ is SOC representable, according to [5], set $\widehat{K}_{\phi,N}^{sc}$ is also SOC representable. $\qquad\square$

We conclude this subsection by making the following remarks about Example 1-Example 3.

– For the generating function in Example 1, if there are $O(M \log(M^3/\sqrt{\epsilon}))$ number of variables and SOC constraints in the representation of set $\widehat{K}_{\phi_1,N}^{sc}$, then we have $K_{\exp}(-\epsilon) \subseteq \widehat{K}_{\phi_1,N}^{sc} \subseteq K_{\exp}(\epsilon)$, where

$$\widehat{K}_{\phi_1,N}^{sc} = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} \displaystyle\sum_{k \in [N]} w_k v_k \geq x_3, \\ r_{1k} = ax_2 - (t_k+1)v_k \geq 0, \forall k \in [N], \\ r_{2k} = a(t_k+1)^{a-1}(x_1 - x_2) - 2^a v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{2^{a+2}(t_k+1)}, r_{1k} - r_{2k}, r_{1k} + r_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{array} \right\}; \qquad (13a)$$

– For the generating function in Example 2, if there are $O(\log\log(M) + \sqrt{\log(M) + \log(1/\epsilon)})$ number of variables and SOC constraints in the representation of set $\widehat{K}_{\phi_2,N}^{sc}$, then we have $K_{\exp}(-\epsilon) \subseteq \widehat{K}_{\phi_2,N}^{sc} \subseteq K_{\exp}(\epsilon)$, where

$$\widehat{K}_{\phi_2,N}^{sc} = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} \displaystyle\sum_{k \in [N]} w_k v_k \geq x_3, \\ r_i \geq 0, \forall i \in [s], \\ (2r_1, x_1 - x_2, x_1 + x_2) \in \mathcal{L}_3, \\ (2r_{i+1}, r_i - x_2, r_i + x_2) \in \mathcal{L}_3, \forall i \in [s-1], \\ \gamma_{1k} = 2^s x_2 - (t_k+1)v_k \geq 0, \forall k \in [N], \\ \gamma_{2k} = r_s - x_2 - 2^{1-s}v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{2^{3-s}(t_k+1)}, \gamma_{1k} - \gamma_{2k}, \gamma_{1k} + \gamma_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{array} \right\}; \qquad (13b)$$

and

– For the generating function in Example 3, if $\widehat{\boldsymbol{x}}$ is a near-optimal solution such that $1-\delta \leq (x_1/x_2)/(\widehat{x}_1/\widehat{x}_2) \leq 1 + \delta$ with $\delta \in (0,1)$ and there are $O(\log(1-\delta)/\log(\delta) + \log(\epsilon)/\log(\delta))$ number of variables and SOC constraints in the representation of set $\widehat{K}_{\phi_3,N}^{sc}$, then we have $K_{\exp}(-\epsilon) \subseteq \widehat{K}_{\phi_3,N}^{sc} \subseteq K_{\exp}(\epsilon)$, where

$$\widehat{K}_{\phi_3,N}^{sc} = \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} \displaystyle\sum_{k \in [N]} w_k v_k \geq x_3/\widehat{x}_2 - (x_2/\widehat{x}_2)\log(\widehat{x}_1/\widehat{x}_2), \\ r_{1k} = x_2/\widehat{x}_2 - (t_k+1)v_k \geq 0, \forall k \in [N], \\ r_{2k} = (x_1/\widehat{x}_1 - x_2/\widehat{x}_2) - 2v_k \geq 0, \forall k \in [N], \\ (v_k\sqrt{8(t_k+1)}, r_{1k} - r_{2k}, r_{1k} + r_{2k}) \in \mathcal{L}_3, \forall k \in [N] \end{array} \right\}. \qquad (13c)$$

3 Approximations Using the Exponential Form (2b)

In this section, we extend the analysis of the approximation result for the exponential function to the exponential cone (2b). We also show that the even-order Taylor approximation of the exponential function is sum-of-squares representable and disprove an SOC representation proposed by [5]. Following the spirit of Example 3 in the previous section, we derive a new shifting approximation scheme using an approximate solution, which overcomes the numerical issues caused by other approximation schemes studied in this section.

3.1 Approximating the Exponential Function and the Exponential Cone

Given the exponential function $\exp(x)$, it is well known that $\lim_{n\to\infty}(1 + x/n)^n = \exp(x)$ for any $x \in \mathbb{R}$. This motivates us to approximate $\exp(x)$ by $(1 + 2^{-N}x)^{2^N}$ for some positive integer $N \in \mathbb{Z}_{++}$. This approximation scheme has been studied in proposition 2.3.7 [5]. However, they did not study the approximation of the exponential cone. This paper fills this gap. We first show the approximation result of the exponential function.

Proposition 1 *Suppose that $x \in [-\widehat{L}, \widehat{L}]$ for some positive parameter $L \in \mathbb{R}_+$ and for any $\epsilon \in (0, \widehat{L})$, let $N = O(\log(\widehat{L}) + \log(1/\epsilon))$. Then we have the following approximation result*

$$\exp(x - \epsilon) \leq \psi_N(x) = (1 + 2^{-N}x)^{2^N} \leq \exp(x).$$

Particularly, the epigraph of $\psi_N(x)$ is SOC representable, i.e.,

$$\text{epi }\psi_N := \left\{ (x, v) \in \mathbb{R} \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N - 1], \\ (2r_1, v - 1, v + 1) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - 1, r_k + 1) \in \mathcal{L}_3, \forall k \in [N - 1], \\ r_N = 1 + 2^{-N}x \end{array} \right\}.$$

Proof: Let $N = \log(4\widehat{L}^2/\epsilon)$. Clearly, we have $2^N > 2\widehat{L}$. Then for $x \in [-\widehat{L}, \widehat{L}]$, setting $y = x/2^N$, we have $|y| \leq 1/2$ and

$$\exp(y - 4y^2) \leq 1 + y \leq \exp(y).$$

Consequently,

$$\exp(2^N y - 2^{N+2}y^2) \leq (1 + y)^{2^N} \leq \exp(2^N y).$$

Since $x = 2^N y$,

$$\exp(x - x^2/2^{N-2}) \leq \left(1 + x/2^N\right)^{2^N} \leq \exp(x).$$

Letting $N = \log(4\widehat{L}^2/\epsilon)$, we have the following approximation bound

$$\exp(x - \epsilon) \leq \left(1 + x/2^N\right)^{2^N} \leq \exp(x), \forall x \in [-\widehat{L}, \widehat{L}].$$

The SOC representation for the epigraph of $\psi_N(x)$ uses the towers of variables and can also be found in [5]. □

Note that this result is different from proposition 2.3.7 [5]. The latter showed $(1 - \epsilon) \exp(x) \leq \psi_N(x) = (1 + 2^{-N}x)^{2^N} \leq \exp(x)$, which is not applicable to the approximation form of the exponential cone (2b).

Since the exponential cone in the form of (2b) can be viewed as the epigraph of perspective of the exponential function, the SOC representation of the approximation in Proposition 1 can be directly applied to the exponential cone (2b).

Theorem 3 *Suppose that $x_3 \in [-2M \log M, 2M \log M]$ and for any $\epsilon \in (0, 2M^2 \log(M))$, let $N = O(\log(M^2 \log M) + \log(1/\epsilon))$ and SOC representable set $\widehat{K}_{\psi,N}^{sc} := \{\boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \mathrm{epi}(\psi_N)\}$. Then, we have $K_{\exp}(0) \subseteq \widehat{K}_{\psi,N}^{sc} \subseteq K_{\exp}(\epsilon)$. Particularly,*

$$\widehat{K}_{\psi,N}^{sc} := \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2r_1, x_1 - x_2, x_1 + x_2) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - x_2, r_k + x_2) \in \mathcal{L}_3, \forall k \in [N-1], \\ r_N = x_2 + 2^{-N} x_3 \end{array} \right\}.$$

Proof: Since $\widehat{K}_{\psi,N}^{sc} := \{\boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \mathrm{epi}(\psi_N)\}$ and $N = O(\log(M^2 \log M) + \log(1/\epsilon))$, the approximation result follows from Proposition 1 by letting $\widehat{L} = 2M^2 \log(M)$. Besides, since set $\mathrm{epi}(\psi_N)$ is SOC representable, according to [5], set $\widehat{K}_{\psi,N}^{sc}$ is also SOC representable. $\qquad\square$

It is worthy of mentioning that the approximation $\widehat{K}_{\psi,N}^{sc}$ obtained in Theorem 3 is an outer one given that $N \geq \log_2(2M^2 \log(M))$.

3.2 Strengthening Using Even-Order Taylor Expansions

The approximation in the previous subsection relies on the fact that $\exp(x) = (\exp(x/2^N))^{2^N}$ and $\exp(x/2^N) \approx 1 + x/2^N$. To strengthen it, we approximate $\exp(x/2^N)$ more tightly using its $2s$-order Taylor expansion with $s \geq 1$. Particularly, letting $y = x/2^N$, we have the following approximation results

$$\exp(y) \approx \widehat{\psi}_{N,2s}(y) := \sum_{i \in [0, 2s]} \frac{y^i}{i!} \text{ and } \exp(x) \approx [\widehat{\psi}_{N,2s}(y)]^{2^N}. \tag{14}$$

We first observe that the polynomial function $\widehat{\psi}_{N,2s}(y) \geq 0$ for all $y \in \mathbb{R}$ and thus is sum-of-squares representable.

Proposition 2 *For any y, the polynomial function $\widehat{\psi}_{N,2s}(y) \geq 0$ and is sum-of-squares representable.*

Proof: We will prove that the minimum of $\widehat{\psi}_{N,2s}(y)$ is nonnegative for any y using induction When $s = 1$, we have $\widehat{\psi}_{N,2}(y) = 1/2 + (y+1)^2/2 > 0$. Suppose $\widehat{\psi}_{N,2t}(y) \geq 0$ for any $s \leq t$. Now let $s = t + 1$. According to induction, we have $d^2\widehat{\psi}_{N,2t+2}/dy^2 = \widehat{\psi}_{N,2t}(y) \geq 0$ and thus $\widehat{\psi}_{N,2t+2}$ is convex. Hence, $\min_y \widehat{\psi}_{N,2s}(y)$ is an unconstrained convex optimization problem. An optimal solution y^* must satisfy the first order optimality condition $d\widehat{\psi}_{N,2t+2}(y^*)/dy = \widehat{\psi}_{N,2t+1}(y^*) = 0$. Thus, we have

$$\min_y \widehat{\psi}_{N,2s}(y) := \widehat{\psi}_{N,2t+2}(y^*) = \widehat{\psi}_{N,2t+1}(y^*) + \frac{(y^*)^{2t+2}}{(2t+2)!} \geq 0.$$

Therefore, the function $\widehat{\psi}_{N,2s}(y) \geq 0$ for any y.

It is known that a nonnegative polynomial function can be written as a sum of squares (see, e.g., [7]). $\square$ Proposition 2 shows that the function $\widehat{\psi}_{N,2s}$ is sum-of-squares representable. However, it is unknown how to represent the function $\widehat{\psi}_{N,2s}$ as an SOC program. In [5], the authors suggested reformulating it as

$$\widehat{\psi}_{N,2s}(y) = \sum_{j \in [0,s]} \frac{\alpha_j}{(2j)!}(\beta_j + y)^{2j} \tag{15}$$

in a hope that $\alpha_j \geq 0$ for all $j \in [0, s]$. We disprove this claim by numerically showing that some components of vector $\boldsymbol{\alpha}$ become negative whenever $s \geq 34$. We first show the following efficient way of finding coefficients of function $\widehat{\psi}_{N,2s}(y)$.

Proposition 3 *The coefficients in* (15) *can be found recursively as*

$$\alpha_s = 1, \beta_s = 1, \sum_{j \in [0,k]} \frac{\alpha_{s-j}\beta_{s-j}^{2k-2j}}{(2k-2j)!} = 1, \sum_{j \in [0,k]} \frac{\alpha_{s-j}\beta_{s-j}^{2k-2j+1}}{(2k-2j+1)!} = 1, \forall k \in [0, s-1]. \tag{16}$$

Proof: See Appendix A.1.

Below, we illustrate some coefficients in Proposition 3 and disprove the claim in [5]

– If $s = 1$, we can reformulate $\widehat{\psi}_{N,2s}(y)$ as

$$\widehat{\psi}_{N,2}(y) = \frac{1}{2} + \frac{1}{2}(y+1)^2$$

– If $s = 2$, we can reformulate $\widehat{\psi}_{N,2s}(y)$ as

$$\widehat{\psi}_{N,4}(y) = \frac{19}{72} + \frac{1}{4}\left(y + \frac{5}{3}\right)^2 + \frac{1}{24}(y+1)^4$$

– If $s = 40$, please find $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in Table 2. We see that $\alpha_0, \ldots, \alpha_6 < 0$. In fact, according to the recursion in Proposition 3, we are unable to represent $\widehat{\psi}_{N,2s}(y)$ in the way of (15) whenever $s \geq 34$.

Although we firmly believe that $\widehat{\psi}_{N,2s}(y)$ is SOC representable for any $s \in \mathbb{Z}_{++}$, we are not able to prove it. Therefore, we leave it as an open question to the interested readers.

Table 2: The values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for $s = 40$. Note: $\alpha_{40} = \beta_{40} = 1$.

k	$\boldsymbol{\alpha}$					k	$\boldsymbol{\beta}$				
0–4	-1.90e-2	-8.09e-3	-2.46e-3	-5.00e-4	-6.16e-5	0–4	37.11	36.86	36.56	36.16	35.59
5–9	-3.88e-6	-8.98e-8	1.17e-10	5.27e-9	1.25e-8	5–9	34.70	33.01	59.20	20.27	19.22
10–14	2.44e-8	4.45e-8	7.94e-8	1.40e-7	2.47e-7	10–14	18.50	17.88	17.29	16.72	16.16
15–19	4.36e-7	7.69e-7	1.36e-6	2.39e-6	4.23e-6	15–19	15.59	15.02	14.45	13.89	13.32
20–24	7.46e-6	1.32e-5	2.33e-5	4.13e-5	7.31e-5	20–24	12.75	12.18	11.61	11.04	10.47
25–29	1.29e-4	2.30e-4	4.08e-4	7.24e-4	1.29e-3	25–29	9.90	9.32	8.75	8.18	7.60
30–34	2.30e-3	4.10e-3	7.34e-3	1.32e-2	2.37e-2	30–34	7.02	6.44	5.86	5.28	4.69
35–39	4.28e-2	7.77e-2	1.42e-1	2.64e-1	5.00e-1	35–39	4.10	3.51	2.91	2.30	1.67

Next, we derive the strengthened approximation results when we use (15) to approximate $\exp(x/2^N)$. The result can be applied to any $s \leq 33$.

Proposition 4 *Suppose that* $x \in [-\widehat{L}, \widehat{L}]$ *for some positive parameter* $\widehat{L} \in \mathbb{R}_+$, $s \in [33]$, *and for any* $\epsilon \in (0, \widehat{L})$, *we let* $N = O(\log(\widehat{L}/s) + \log(1/\epsilon)/s)$. *Then we have the following approximation result*

$$\exp(x - \epsilon) \leq \overline{\psi}_{N,2s}(x) = (\widehat{\psi}_{N,2s}(x/2^N))^{2^N} \leq \exp(x + \epsilon).$$

Particularly, when $s = 1$, *the epigraph of* $\overline{\psi}_{N,2}(x)$ *admits the following SOC representation:*

$$\text{epi}\,\overline{\psi}_{N,2} := \left\{ (x, v) \in \mathbb{R} \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N], \\ (2r_1, v - 1, v + 1) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - 1, r_k + 1) \in \mathcal{L}_3, \forall k \in [N-1], \\ (1 + 2^{-N}x, r_N - 1, r_N) \in \mathcal{L}_3 \end{array} \right\}.$$

Proof: For notational convenience, we let $y = x/2^N$. Suppose that $N \geq \log_2(\widehat{L})$. Then we have $|y| \leq 1$.

Next, we prove the approximation result by discussing whether $y \geq 0$ or not.

Case 1. If $y \geq 0$, according to the error formula of Taylor approximation, we have

$$0 \leq \exp(y) - \widehat{\psi}_{N,2s}(y) \leq \frac{y^{2s+1}}{(2s+1)!} \exp(y).$$

According to the proof of Proposition 1, we have

$$\exp\left(y - \frac{y^{2s+1}}{(2s+1)!} - \frac{4y^{4s+2}}{((2s+1)!)^2}\right) \leq \widehat{\psi}_{N,2s}(y) \leq \exp(y).$$

Since $x = 2^N y$, we obtain

$$\exp\left(x - \frac{x^{2s+1}}{2^{2Ns}(2s+1)!} - \frac{x^{4s+2}}{2^{N(4s+1)-2}((2s+1)!)^2}\right) \leq \overline{\psi}_{N,2s}(x) \leq \exp(x).$$

Using the facts that $|x| \leq \widehat{L}$, $N \geq 2$, and $(2s+1)! \geq ((2s+1)/e)^{2s+1}$, by choosing

$$N \geq \frac{(2s+1)\log(e\widehat{L}/(2s+1)) + \log(1/\epsilon)}{2s\log(2)}$$

or $N = O(\log(\widehat{L}/s) + \log(1/\epsilon)/s)$, we have

$$\exp(x - \epsilon) \leq \overline{\psi}_{N,2s}(x) \leq \exp(x + \epsilon), \forall x \in [0, \widehat{L}].$$

Case 2. Similarly, if $y \leq 0$, according to the error formula of Taylor approximation, we have

$$0 \geq \exp(y) - \widehat{\psi}_{N,2s}(y) \geq \frac{y^{2s+1}}{(2s+1)!}.$$

As $\exp(y) \geq y^2/3$ for any $y \in [-1, 0]$, we have

$$\exp\left(y - \frac{3y^{2s-1}}{(2s+1)!}\right) \geq \widehat{\psi}_{N,2s}(y) \geq \exp(y).$$

Since $x = 2^N y$, we obtain

$$\exp\left(x - \frac{3x^{2s-1}}{2^{N(2s-2)}(2s+1)!}\right) \geq \overline{\psi}_{N,2s}(x) \geq \exp(x).$$

Similarly, by choosing $N = O(\log(\widehat{L}/s) + \log(1/\epsilon)/s)$, we have

$$\exp(x - \epsilon) \leq \overline{\psi}_{N,2s}(x) \leq \exp(x + \epsilon), \forall x \in [-\widehat{L}, 0].$$

This proves the approximation result.

The SOC representation for the epigraph of $\overline{\psi}_{N,2}(x)$ follows from Proposition 1 and Proposition 3. □
Note that if the conjecture that $\widehat{\psi}_{N,2s}(y)$ is SOC representable were true for any $s \in \mathbb{Z}_{++}$, then one could choose $N = s = O(\sqrt{\log(1/\epsilon)})$ given that $\log(\widehat{L})$ is a constant. This matches the best approximation result in Corollary 2 of Example 2. In our numerical study, we use $s = 1$ (i.e., set $\overline{\psi}_{N,2}(x)$) to effectively solve MIECPs, which unfortunately still has a numerical issue that cannot close the gap even if N is very large. We will use the shifting method in the next subsection to resolve it.

Since the exponential cone in the form of (2b) can be viewed as the epigraph of perspective of the exponential function, the approximation result, as well as the SOC representations in Proposition 4, can be directly applicable to the exponential cone (2b).

Theorem 4 *Suppose that $x_3 \in [-2M \log M, 2M \log M]$, $s \in [33]$, and for $\epsilon \in (0, 2M^2 \log(M))$, let $N = O(\log(M^2 \log(M)/s) + \log(1/\epsilon)/s)$. Then we have conic representable $\widehat{K}^{sc}_{\overline{\psi}, N, 2s} := \{x \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{epi}(\overline{\psi}_{N,2s})\}$ such that $K_{\exp}(-\epsilon) \subseteq \widehat{K}^{sc}_{\overline{\psi}, N, 2s} \subseteq K_{\exp}(\epsilon)$. Particularly, when $s = 1$, we have*

$$\widehat{K}^{sc}_{\overline{\psi}, N, 2} := \left\{ x \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2r_1, x_1 - x_2, x_1 + x_2) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - x_2, r_k + x_2) \in \mathcal{L}_3, \forall k \in [N-1], \\ (x_2 + 2^{-N} x_3, r_N - x_2, r_N) \in \mathcal{L}_3 \end{array} \right\}.$$

Proof: Since $\widehat{K}^{sc}_{\overline{\psi}, N, 2s} := \{x \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{epi}(\overline{\psi}_{N,2s})\}$ and $N = O(\log(M^2 \log(M)/s) + \log(1/\epsilon)/s)$, the approximation result follows from Proposition 4 by letting $\widehat{L} = 2M^2 \log(M)$. Besides, since set $\text{epi}(\overline{\psi}_{N,2s})$ is SOC representable, according to [5], set $\widehat{K}^{sc}_{\overline{\psi}, N, 2s}$ is also SOC representable. $\square$

3.3 Shifting Using Approximate Solutions

Similar to Example 3, we propose to extend the approximation schemes in the previous subsections using the shifting method provided that an approximate solution is available. To begin with, we summarize the results for applying the shifting method to the approximation proposed in Proposition 1.

Proposition 5 *Suppose $\widehat{x}$ is a given approximate solution such that $|x - \widehat{x}| \leq \delta$ with $\delta > 0$ and for any $\epsilon \in (0, \delta]$, let $N = O(\log(\delta^2/\epsilon))$. Then we have the following approximation bound*

$$\exp(x - \epsilon) \leq \exp(\widehat{x}) \psi_N^{sft}(x) = \exp(\widehat{x})(1 + 2^{-N}(x - \widehat{x}))^{2^N} \leq \exp(x).$$

Particularly, the epigraph of $\psi_N^{sft}(x)$ is SOC representable, i.e.,

$$\text{epi}\,\psi_N^{sft} := \left\{ (x, v) \in \mathbb{R} \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2r_1, v - 1, v + 1) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - 1, r_k + 1) \in \mathcal{L}_3, \forall k \in [N-1], \\ r_N = 1 + 2^{-N}(x - \widehat{x}) \end{array} \right\}.$$

Proof: The proof follows directly from that of Proposition 1 by letting $y = (x - \widehat{x})/2^N$. $\square$

Similarly, since the exponential cone in the form of (2b) can be viewed as the epigraph of perspective of the exponential function, the shifting results in Proposition 5 can be directly applicable to the exponential cone (2b).

Theorem 5 *Suppose $\widehat{x}$ is an approximate solution such that $|x_3/x_2 - \widehat{x}_3/\widehat{x}_2| \leq \delta$ with $\delta > 0$ and for any $\epsilon \in (0, \delta]$, let $N = O(\log(\delta^2/\epsilon))$ and set $\widehat{K}^{sc,sft}_{\psi, N} := \{x \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{epi}(\psi_N^{sft})\}$. Then we have $K_{\exp}(0) \subseteq \widehat{K}^{sc,sft}_{\psi, N} \subseteq K_{\exp}(\epsilon)$. Particularly,*

$$\widehat{K}^{sc,sft}_{\psi, N} := \left\{ x \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2\sqrt{\exp(\widehat{x}_3/\widehat{x}_2)}\, r_1, x_1 - x_2, x_1 + x_2) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - x_2, r_k + x_2) \in \mathcal{L}_3, \forall k \in [N-1], \\ r_N = x_2 + 2^{-N}(x_3 - x_2(\widehat{x}_3/\widehat{x}_2)) \end{array} \right\}.$$

Proof: The proof is similar to that of Theorem 3 and thus is omitted. $\square$

Next, we summarize the results for applying the shifting method to the approximation scheme proposed in Section 3.2.

Proposition 6 *Suppose $\widehat{x}$ is an approximate solution such that $|x - \widehat{x}| \leq \delta$ with $\delta > 0$ and for any $\epsilon \in (0, \delta]$, let $N = O(\log(\delta/s) + \log(1/\epsilon)/s)$. Then we have the following approximation result*

$$\exp(x - \epsilon) \leq \exp(\widehat{x})\overline{\psi}_{N,2s}^{sft}(x) = \exp(\widehat{x})(\widehat{\psi}_{N,2s}((x-\widehat{x})/2^N))^{2^N} \leq \exp(x + \epsilon).$$

Particularly, when $s = 1$, the epigraph of $\overline{\psi}_{N,2}^{sft}(x)$ is SOC representable, i.e.,

$$\text{epi}(\overline{\psi}_{N,2}^{sft}) := \left\{ (x, v) \in \mathbb{R} \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2r_1, v-1, v+1) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - 1, r_k + 1) \in \mathcal{L}_3, \forall k \in [N-1], \\ (1 + 2^{-N}(x - \widehat{x}), r_N - 1, r_N) \in \mathcal{L}_3 \end{array} \right\}.$$

Proof: The proof follows from Proposition 4 by letting $y = (x - \widehat{x})/2^N$. □

Theorem 6 *Suppose $\widehat{\boldsymbol{x}}$ is an approximate solution such that $|x_3/x_2 - \widehat{x}_3/\widehat{x}_2| \leq \delta$ with $\delta > 0$ and for any $\epsilon \in (0, \delta]$, let $N = O(\log(\delta/s) + \log(1/\epsilon)/s)$ and set $\widehat{K}_{\overline{\psi},N,2s}^{sc,sft} := \{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \text{epi}(\overline{\psi}_{N,2s}^{sft}) \}$ such that $K_{\exp}(-\epsilon) \subseteq \widehat{K}_{\overline{\psi},N,2s}^{sc,sft} \subseteq K_{\exp}(\epsilon)$. Particularly, when $s = 1$, we have*

$$\widehat{K}_{\overline{\psi},N,2}^{sc,sft} := \left\{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : \begin{array}{l} r_k \geq 0, \forall k \in [N-1], \\ (2r_1\sqrt{\exp(\widehat{x}_3/\widehat{x}_2)}, x_1 - x_2, x_1 + x_2) \in \mathcal{L}_3, \\ (2r_{k+1}, r_k - x_2, r_k + x_2) \in \mathcal{L}_3, \forall k \in [N-1], \\ (x_2 + 2^{-N}(x_3 - x_2(\widehat{x}_3/\widehat{x}_2)), r_N - x_2, r_N) \in \mathcal{L}_3 \end{array} \right\}.$$

Proof: The proof is similar to that of Theorem 4 and thus is omitted. □

We remark that (i) the number of variables and SOC constraint (i.e., N) depends on the approximation accuracy ϵ and the quality of a given approximate solution. Thus, a better approximate solution reduces N; and (ii) in practice, to solve MIECP, there may exist many heuristic methods, and leveraging their near-optimal solutions can help us find an optimal solution more effectively. This phenomenon has been observed in our numerical study.

3.4 Comparisons of Methods in Section 3

The numerical comparison of Section 3.1, Section 3.2, Section 3.1 Shift, and Section 3.2 Shift methods to approximate the exponential function can be found in Figure 2, where we let $\widehat{x} = 1$ for the shifting methods. It is seen that the shifting methods can improve both Section 3.1 and Section 3.2 methods. In practice, we can run heuristics to obtain a good-quality $\widehat{x}$.

4 Lower Bounds for Polyhedral and SOC Approximations in the Extended Space

This section focuses on deriving the minimum size (i.e., the minimum number of variables and constraints) of an extended polyhedron or an ex-

Fig. 2: A Comparison of Methods in Section 3 using Two SOC Constraints to Approximate the Exponential Function in the Domain $[0.3, 3.0]$

tended SOC program to approximate the exponential cone within an approximation accuracy.

To begin with, we first study the minimum size of an extended polyhedron P_q with q number of variables and constraints to approximate the hypograph of $\log(x)$ with approximation accuracy ϵ.
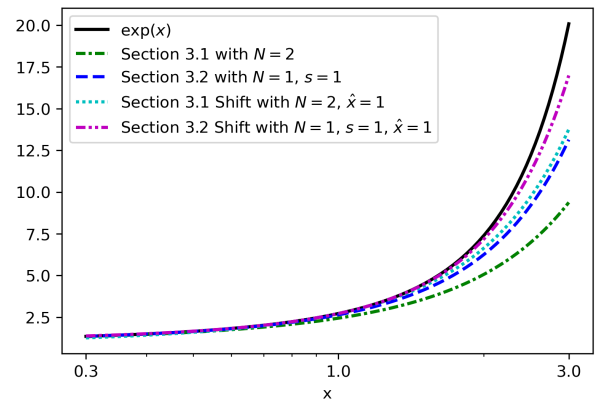
Theorem 7 *For any $\epsilon \in (0,1]$, any extended polyhedron P_q with q variables (including variables x, ν defined in (6)) and constraints such that $\mathrm{hyp}\log(-\epsilon) \subseteq \mathrm{Proj}_{(x,\nu)}(P_q) \subseteq \mathrm{hyp}\log(\epsilon)$. Then we must have $q = \Omega(\log\log(M) + \log(1/\epsilon))$.*

Proof: Any extended polyhedron P_q with q number of variables and constraints has at most 2^q extreme points so is its projection $\mathrm{Proj}_{(x,\nu)}(P_q)$. Now let us consider the two extreme points of $\mathrm{Proj}_{(x,\nu)}(P_q)$, denoted by $(\bar{x}_1, \log(\bar{x}_1)+\delta_1), (\bar{x}_2, \log(\bar{x}_2)+\delta_2)$, which are consecutive in x-space such that $\bar{x}_1 < \bar{x}_2$. Since $\mathrm{hyp}\log(\epsilon) \subseteq \mathrm{Proj}_{(x,\nu)}(P_q) \subseteq \mathrm{hyp}\log(-\epsilon)$, we have

$$|\delta_1| \le \epsilon, |\delta_2| \le \epsilon$$

Then, the largest distance in the ν-direction between $\mathrm{Proj}_{(x,\nu)}(P_q) \cap \{(x,v) : x \in [\bar{x}_1, \bar{x}_2]\}$ and set $\mathrm{hyp}\log(\epsilon) \cap \{(x,v) : x \in [\bar{x}_1, \bar{x}_2]\}$ can be lower bounded as

$$d_H = \min_{\delta_1 \in [-\epsilon,\epsilon], \delta_2 \in [-\epsilon,\epsilon]} \max_{x \in [\bar{x}_1, \bar{x}_2]} \left| \log(x) - \frac{\log(\bar{x}_2) + \delta_2 - \log(\bar{x}_1) - \delta_1}{\bar{x}_2 - \bar{x}_1}(x - \bar{x}_1) - \log(\bar{x}_1) - \delta_1 \right|. \tag{17a}$$

In (17a), without loss of generality, we can define $x := x/\bar{x}_1$ and $t = \bar{x}_2/\bar{x}_1 > 1$. Thus, the optimization problem (17a) can be simplified as

$$d_H = \min_{\delta_1 \in [-\epsilon,\epsilon], \delta_2 \in [-\epsilon,\epsilon]} \max_{x \in [1,t]} \left| \log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t - 1}(x - 1) - \delta_1 \right|. \tag{17b}$$

Note that to achieve the outer minimum in (17b), there are only four valid cases: (a) $\epsilon \ge \delta_1 \ge 0 \ge \delta_2 \ge -\epsilon$; (b) $\epsilon \ge \delta_2 \ge 0 \ge \delta_1 \ge -\epsilon$; (c) $\epsilon \ge \delta_1, \delta_2 \ge 0$; and (d) $0 \ge \delta_1, \delta_2 \ge -\epsilon$. Please see Figure 3 for an illustration.

Case (a). If $\epsilon \ge \delta_1 \ge 0 \ge \delta_2 \ge -\epsilon$, then (17b) can be simplified as

$$d_{H1} = \min_{\epsilon \ge \delta_1 \ge 0 \ge \delta_2 \ge -\epsilon} \max \left\{ \delta_1, -\delta_2, \max_{x \in [1,t]} \log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t - 1}(x - 1) - \delta_1 \right\}.$$

Taking the average of the first and third expressions in the first maximization operator, the value d_{H1} can be lower bounded by

$$d_{H1} \ge \min_{\epsilon \ge \delta_1 \ge 0 \ge \delta_2 \ge -\epsilon} \frac{1}{2} \max_{x \in [1,t]} \left[\log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t - 1}(x - 1) \right]$$

$$= \frac{1}{2} \max_{x \in [1,t]} \left[\log(x) - \frac{\log(t)}{t - 1}(x - 1) \right]$$

$$= \frac{1}{2} \left[-\log\left(\frac{\log(t)}{t - 1} \right) - 1 + \frac{\log(t)}{t - 1} \right] \tag{17c}$$

where the first equality is due to the monotonicity of the objective function with respect to δ_1, δ_2, and the second one is because of the optimality condition and concavity.

Case (b). If $\epsilon \ge \delta_2 \ge 0 \ge \delta_1 \ge -\epsilon$, then (17b) can be simplified as

$$d_{H2} = \min_{\epsilon \ge \delta_2 \ge 0 \ge \delta_1 \ge -\epsilon} \max \left\{ -\delta_1, \delta_2, \max_{x \in [1,t]} \log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t - 1}(x - 1) - \delta_1 \right\}.$$

Taking the average of the second and third expressions in the first maximization operator, the value d_{H2} can be lower bounded by

$$d_{H2} \ge \min_{\epsilon \ge \delta_1 \ge 0 \ge \delta_2 \ge -\epsilon} \frac{1}{2} \max_{x \in [1,t]} \left[\log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t - 1}(x - 1) - \delta_1 + \delta_2 \right]$$

$$= \frac{1}{2} \max_{x \in [1,t]} \left[\log(x) - \frac{\log(t)}{t - 1}(x - 1) \right]$$

(a) $\epsilon \geq \delta_1 \geq 0 \geq \delta_2 \geq -\epsilon$

(b) $\epsilon \geq \delta_2 \geq 0 \geq \delta_1 \geq -\epsilon$

(c) $\epsilon \geq \delta_1, \delta_2 \geq 0$

(d) $0 \geq \delta_1, \delta_2 \geq -\epsilon$
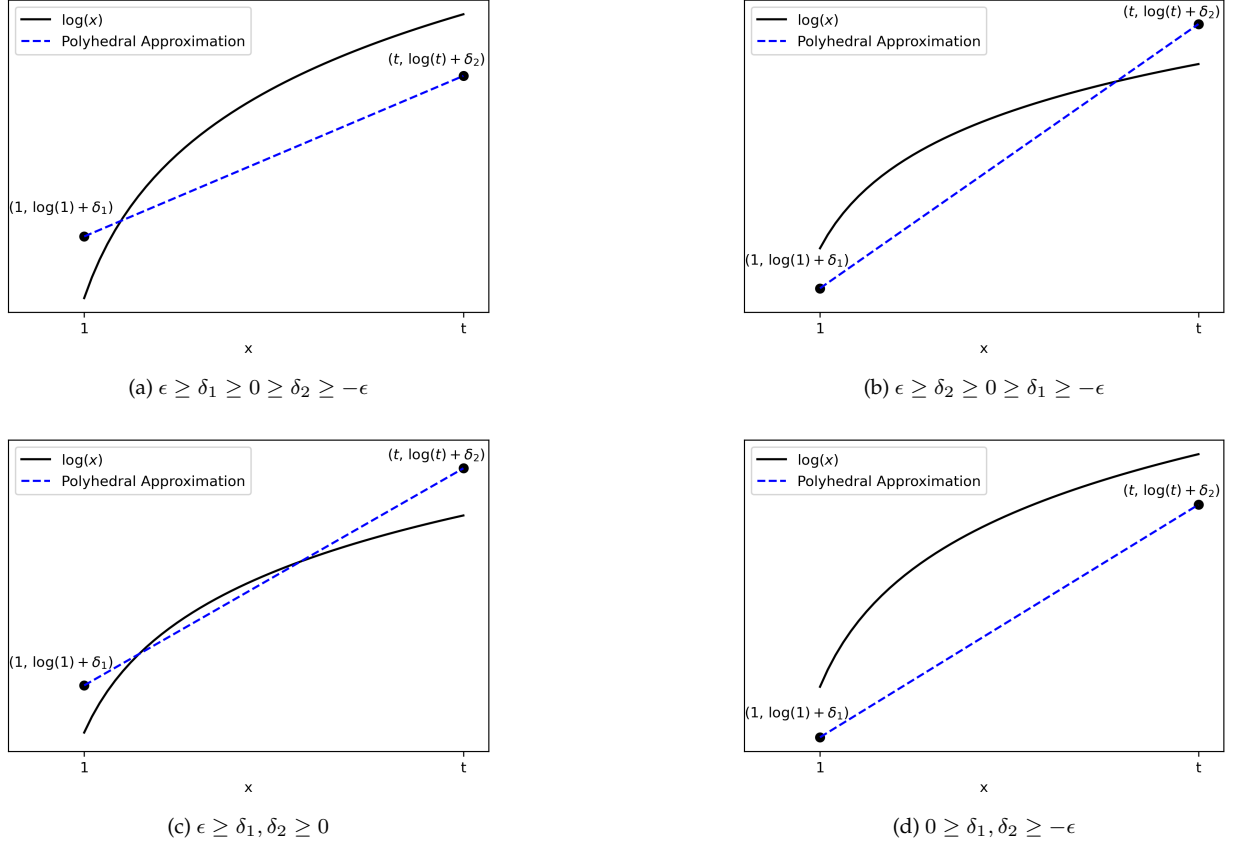
Fig. 3: Four Cases for the Lower Bound of Polyhedral Approximation

$$= \frac{1}{2}\left[-\log\left(\frac{\log(t)}{t-1}\right) - 1 + \frac{\log(t)}{t-1}\right] \tag{17d}$$

where the first equality is due to monotonicity of the objective function with respect to δ_1, δ_2, and the second one is because of the optimality condition and concavity.

Case (c). If $\epsilon \geq \delta_1, \delta_2 \geq 0$, then (17b) can be simplified as

$$d_{H3} = \min_{\epsilon \geq \delta_1,\delta_2 \geq 0} \max\left\{\delta_1, \delta_2, \max_{x\in[1,t]} \log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t-1}(x-1) - \delta_1\right\}.$$

By symmetry, we must have $\delta_1 = \delta_2$ at optimality. Thus, we have

$$d_{H3} = \min_{\epsilon \geq \delta_1 \geq 0} \max\left\{\delta_1, \max_{x\in[1,t]} \log(x) - \frac{\log(t)}{t-1}(x-1) - \delta_1\right\}.$$

Taking average of the first and second expressions in the first maximization operator, the value d_{H3} can be lower bounded by

$$d_{H3} \geq \frac{1}{2}\max_{x\in[1,t]}\left[\log(x) - \frac{\log(t)}{t-1}(x-1)\right] = \frac{1}{2}\left[-\log\left(\frac{\log(t)}{t-1}\right) - 1 + \frac{\log(t)}{t-1}\right] \tag{17e}$$

where the first equality is because of the optimality condition and concavity.

Case (d). If $0 \geq \delta_1, \delta_2 \geq -\epsilon$, then (17b) can be simplified as

$$d_{H4} = \min_{0 \geq \delta_1, \delta_2 \geq -\epsilon} \max\left\{ -\delta_1, -\delta_2, \max_{x \in [1,t]} \log(x) - \frac{\log(t) + \delta_2 - \delta_1}{t-1}(x-1) - \delta_1 \right\}.$$

Since the first maximization is monotone non-increasing with respect to δ_1, δ_2, thus the outer minimization is achieved by $\delta_1 = 0, \delta_2 = 0$. Then we have

$$d_{H4} = \max_{x \in [1,t]} \log(x) - \frac{\log(t)}{t-1}(x-1) = -\log\left(\frac{\log(t)}{t-1}\right) - 1 + \frac{\log(t)}{t-1} \tag{17f}$$

where the second equality is because of the optimality condition and concavity.

Thus, we have

$$d_H = \min_{i \in [4]}\{d_{Hi}\} \geq \frac{1}{2}\left[-\log\left(\frac{\log(t)}{t-1}\right) - 1 + \frac{\log(t)}{t-1} \right] \geq \frac{1}{10^4}(t-1)^2$$

for any $t > 1$ and $d_H \leq \epsilon \leq 1$. Hence, to achieve $\mathrm{hyp}\log(\epsilon) \subseteq \mathrm{Proj}_{(x,\nu)}(P_q) \subseteq \mathrm{hyp}\log(-\epsilon)$, we must have $10^{-4}(t-1)^2 \leq \epsilon$, i.e., $t = \bar{x}_2/\bar{x}_1 \leq 1 + \sqrt{10^4 \epsilon}$.

Since $(\bar{x}_1, \log(\bar{x}_1) + \delta_1), (\bar{x}_2, \log(\bar{x}_2) + \delta_2)$ are two arbitrarily consecutive extreme points, for all the extreme points, we must have

$$(1 + \sqrt{10^4 \epsilon})^{2^q} \geq M/(1/M) := M^2$$

i.e., $q = \Omega(\log\log(M) + \log(1/\epsilon))$. $\qquad\square$

We notice that the exponential cone can be viewed as the hypograph of the perspective of the logarithm function. Thus, the lower bound result in Theorem 7 can be directly applied to that of polyhedral approximation of the exponential cone.

Corollary 4 *For any $\epsilon \in (0,1]$, given a polyhedral approximation P_q with q additional variables and linear constraints such that $K_{\exp}(-\epsilon) \subseteq P_q \subseteq K_{\exp}(\epsilon)$, we must have $q = \Omega(\log\log(M) + \log(1/\epsilon))$.*

In [6], the authors showed a lower bound of any polyhedral approximation of an SOC program is $\Omega(\log(1/\epsilon))$. Together with the lower bound result in Theorem 7, we have that to obtain an ϵ- SOC approximation of the exponential cone, it requires $\Omega(1 + \log\log(M)/\log(1/\epsilon))$ number of variables and SOC constraints. This lower bound result is summarized below.

Corollary 5 *For any $\epsilon \in (0,1]$, given an SOC approximation C_q with q number of variables and SOC constraints such that $K_{\exp}(-\epsilon) \subseteq C_q \subseteq K_{\exp}(\epsilon)$, we have $q = \Omega(1 + \log\log(M)/\log(1/\epsilon))$.*

This corollary shows that the minimum number of SOC constraints needed to approximate the exponential cone may be constant, given that $\log(M)$ is constant. Meanwhile, according to Section 2.2, if there exists a tight approximate solution, then we can also obtain a constant number of variables and SOC constraints using Example 3 to achieve $\epsilon-$approximation accuracy; otherwise, according to the discussion of Example 2 in Section 2.2 or Theorem 4 with $N = s$, the best we can achieve is $O(\sqrt{\log(1/\epsilon)})$ number of additional variables and second-order constraints. However, we are unable to further improve the lower or upper bounds and leave them to interested readers.

5 Approximating the Exponential Cone in the Original Space Using Gradient Inequalities

This section will study the polyhedral outer approximation of the exponential cone in the original space based on gradient inequalities. We study the upper and lower bounds for the approximation errors.

We first study the outer approximation of the hypograph of $\log(x)$, i.e., $\mathrm{hyp}\log(0)$. Given N points $\{\widehat{x}_i\}_{i\in[N]}$ (including two boundary points of $\mathcal{R}$ for the sake of simplicity) in the approximation, the outer approximation, denoted by $\widehat{H}_N^o$, admits the following form

$$\widehat{H}_N^o = \left\{ (x,\nu) \in \mathbb{R} \times \mathbb{R} : \log(\widehat{x}_i) + \frac{1}{\widehat{x}_i}(x - \widehat{x}_i) \geq \nu \right\}. \tag{18}$$

Since $\log(x)$ is a concave function, we have $\widehat{H}_N^o \subseteq \mathrm{hyp}\log(0)$. We observe that the approximation error of polyhedral outer approximation depends on the maximum distance between the extreme points and logarithm function. Based on this observation, we can find a tight approximation bound in the ν direction, and the result is summarized below.

Theorem 8 *For any $\epsilon \in [0,1]$, there exists a polyhedral outer approximation in the original space with $N = \Theta(\log(M) + 1/\sqrt{\epsilon})$ points such that $\mathrm{hyp}\log(0) \subseteq \widehat{H}_N^o \subseteq \mathrm{hyp}\log(\epsilon)$.*

Proof: We observe that except the boundary points, there are $N-1$ extreme points in set $\widehat{H}_N^o$. Now let us consider its two neighboring points $(\bar{x}_1, \log(\bar{x}_1)), (\bar{x}_2, \log(\bar{x}_2))$ which are consecutive in x-space and satisfy $\bar{x}_1 < \bar{x}_2$. Then, the largest distance on the ν-direction between $\widehat{H}_N^o \cap \{(x,\nu) : x \in [\bar{x}_1, \bar{x}_2]\}$ and set $\mathrm{hyp}\log(0) \cap \{(x,\nu) : x \in [\bar{x}_1, \bar{x}_2]\}$ is

$$d = \frac{1}{\bar{x}_1}(x^* - \bar{x}_1) + \log(\bar{x}_1) - \log(x^*), \tag{19}$$

where $x^* = \log(\bar{x}_2/\bar{x}_1)/(1/\bar{x}_1 - 1/\bar{x}_2)$ is the x-coordinate of the intersection point of two lines $y = (x - \bar{x}_1)/\bar{x}_1 + \log(\bar{x}_1)$ and $(x - \bar{x}_2)/\bar{x}_2 + \log(\bar{x}_2)$. Please see Figure 4 for an illustration.

To simplify (19), we define $t = \bar{x}_2/\bar{x}_1 > 1$. Thus, the distance formula (19) can be simplified as

$$d = -\log\left(\frac{\log(t)}{t-1}\right) - 1 + \frac{\log(t)}{t-1} \leq \frac{1}{8}(t-1)^2,$$

and the inequality holds for any $t > 1$. Hence, to achieve $\mathrm{hyp}\log(-\epsilon) \subseteq \widehat{H}_N \subseteq \mathrm{hyp}\log(0)$, it suffices to let $t = \bar{x}_2/\bar{x}_1 \leq 1 + \sqrt{8\epsilon}$. Since $(\bar{x}_1, \log(\bar{x}_1)), (\bar{x}_2, \log(\bar{x}_2))$ are two arbitrarily consecutive points, let us choose set $\{\bar{x}_i\}_{i\in[N]}$ as $\widehat{x}_{i+1}/\widehat{x}_i = 1 + \sqrt{8\epsilon}$ for all $i \in [N-1]$. Then we must have

$$(1 + \sqrt{8\epsilon})^{N-1} \geq M/(1/M) := M^2$$

Fig. 4: Illustration of the Polyhedral Outer Approximation Error

i.e., $N = O(\log(M) + 1/\sqrt{\epsilon})$. Following the proof in Theorem 7 by replacing 2^q by N, we can obtain that the smallest $N = \Omega(\log(M) + 1/\sqrt{\epsilon})$. Thus, we must have $N = \Theta(\log(M) + 1/\sqrt{\epsilon})$. $\square$

Given that $\log(M)$ is constant, Theorem 8 shows the best outer approximations in the original space that can achieve ϵ-approximation accuracy require $N = O(1/\sqrt{\epsilon})$. Different from the results in the other sections, $N = O(1/\sqrt{\epsilon})$ can be significantly larger than $N = O(\log(1/\epsilon))$ or $N = O(\sqrt{\log(1/\epsilon)})$. However, in our numerical study, we find that the outer approximations work very well especially having mixed-integer variables, which may be because the solvers are better at solving MILPs compared to MISOCPs.

Since the exponential cone can be viewed as the hypograph of perspective of the logarithm function, the approximation result also holds for the exponential cone. The upper and lower bounds results in Theorem 8 can be directly applied to the polyhedral outer approximation of the exponential cone.
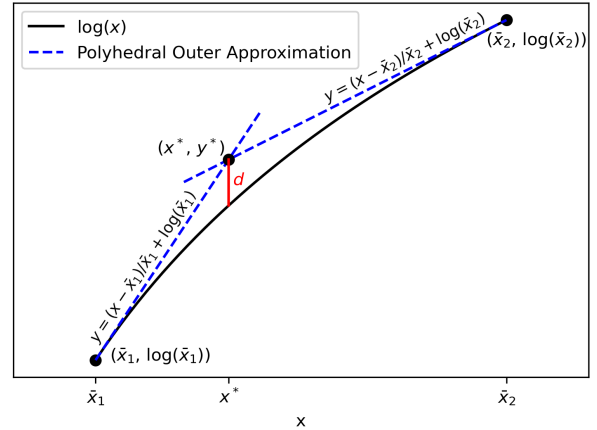
Corollary 6 *For any $\epsilon \in [0, 1]$, there exists a polyhedral outer approximation in the original space with $\Theta(\log(M) + 1/\sqrt{\epsilon})$ points such that $K_{\exp}(0) \subseteq \widehat{K}_N^o \subseteq K_{\exp}(\epsilon)$, where $\widehat{K}_N^o := \{ \boldsymbol{x} \in \mathcal{R}^2 \times \mathbb{R} : (x_1/x_2, x_3/x_2) \in \widehat{H}_N^o \}$.*

In practice, different from SOC approximations, the proposed outer approximation can be done more efficiently in an iterative way. That is, given a solution $\widehat{\boldsymbol{x}} \in \mathbb{R}^2 \times \mathbb{R} \setminus K_{\exp}(0)$, i.e., $(\widehat{x}_1/\widehat{x}_2, \widehat{x}_3/\widehat{x}_2) \notin \operatorname{hyp}\log(0)$. Then we can add a gradient inequality to cut it off, i.e.,

$$\log(\widehat{x}_1/\widehat{x}_2) + \widehat{x}_2/\widehat{x}_1(x_1/x_2 - \widehat{x}_1/\widehat{x}_2) \geq x_3/x_2,$$

which is equivalent to

$$x_2 \log(\widehat{x}_1/\widehat{x}_2) + x_1\widehat{x}_2/\widehat{x}_1 - x_2 \geq x_3.$$

We proceed until reaching a stopping criterion (e.g., the relative gap is within 10^{-4}).

We plan to employ the naive cutting plane method and the branch and cut algorithm to solve the MIECP, where the latter generates a gradient inequality via a delayed cut generating procedure. Our numerical study shows that the proposed cutting plane and branch and cut methods based on gradient inequalities can effectively solve MIECPs.

6 Numerical Study

In this section, we provide numerical illustrations for both SOC approximations and polyhedral approximations by solving three types of MIECPs, i.e., a packing MIECP, a covering MIECP, and a sparse logistic regression problem. The proposed methods are implemented in Gurobi, and we compare the results to the state-of-art solver MOSEK. All the instances in this section are executed in Python 3.7 with calls to Gurobi 9.0 and MOSEK 9.2 on a personal computer with 2.3 GHz Intel Core i9 processor and 16G of memory. Codes of the numerical experiments are available at `https://github.com/qingye1/Exp_Cone_Approximation`.

Nomenclature for the Numerical Study: The optimality gap, denoted as "Gap", is computed based on the true optimal solution of an MIECP, which is computed by the absolute difference between the best objective value of an approximation method and the true optimal value of an MIECP divided by the true optimal value. We use dash line "—" to mark the cases whose Gaps are no larger than 10^{-4} and use asterisk "*" to mark cases that exceed the time limit. We also compute the ratio of the running time of a proposed method to that of MOSEK, denoted as "Ratio," and use the geometric mean of ratios to evaluate computational efficiency, denoted as "Geo Mean." We use "Section 3.1" and "Section 3.2" to denote the methods based on the SOC approximation results in Section 3.1 and Section 3.2, and use "Section 3.1 Shift" and "Section 3.2 Shift" to denote their corresponding shifting methods studied in Section 3.3. We use "Cutting Plane" to denote naive implementation of cutting plane method without any delayed implementations and "Branch and Cut" to denote the branch and cut algorithm with delayed constraint generations. When employing Example 3, we select the approximate solution from the list $\{2^{-6}, 2^{-5}, \ldots, 2^0\}$. In "Example 3 with Best Scale", we construct the best approximate solution for each exponential conic constraint using a feasible solution found by executing Example 3 with 3-tuple (N, AS, TL), where $N \in [4]$, and time limit $TL \in \{2, 10, 30\}$ (to ensure a feasible solution to be found), and an approximate solution AS is chosen from the list $\{2^{-6}, 2^{-5}, \ldots, 2^0\}$. This procedure is called "Best Scale." We follow the same procedure to find the best approximate solution for the shifting methods in Section 3.1 and Section 3.2. Note that we report the total time for constructing the best approximate solution using "Best Scale" and solving an MIECP for methods employing the "Best Scale" procedure. We use "n/a" to denote that an instance cannot be solved by one method. We call a method running into "numerical issues" if we cannot close the gap by increasing the number of SOC or polyhedral constraints.

6.1 Packing MIECP

Let us consider the following mixed-integer convex packing problem

$$\min_{\boldsymbol{x} \in X} \left\{ \sum_{\ell \in [p]} \exp \left(\sum_{j \in [n]} c_{\ell j} x_j \right) : \sum_{j \in [n]} a_{ij} x_j \leq b_i, \forall i \in [m] \right\}, \tag{20}$$

where $X = \{0, 1\}^t \times [0, 1]^{n-t}$. Note that the problem (20) can be formulated as the following packing MIECP

$$\min_{\boldsymbol{x} \in X} \left\{ \sum_{\ell \in [p]} v_\ell : \left(v_\ell, 1, \sum_{j \in [n]} c_{\ell j} x_j \right) \in K_{\exp}(0), \forall \ell \in [p], \sum_{j \in [n]} a_{ij} x_j \leq b_i, \forall i \in [m] \right\}. \tag{21}$$

In this subsection, we test the proposed methods by solving the packing MIECP (21). We conduct two experiments to test the proposed SOC and polyhedral approximations for solving (21), where we consider pure binary ($t = n$) and mixed-integer ($t \in [n-1]$) experiments and compare our results with MOSEK. In the testing instances, we set $m = 100$ and suppose $a_{ij} \sim \text{int}(0, 9)$, $b_i = 4n$, $c_{\ell j} \sim -\text{int}(0, 9)/n$ for all $i \in [m], j \in [n], \ell \in [p]$, where $\text{int}(p, q)$ denotes a random integer between p and q including p and q. The time limit is set to be 3600 seconds for both experiments. Due to different experimental settings (e.g., the accuracy requirements), the pure continuous ($t = 0$) experiment is reported in Appendix C.

Experiment 1- Binary Packing MIECP: For the binary experiment, we solve the small-scale instances with $n = 20$, $p \in \{5, 7, \ldots, 25\}$ to do sensitive analyses of parameters and learn the patterns and then solve the large-scale problem with $n = 100$, $p \in \{5, 10, \ldots, 50\}$. Due to the page limit, we do not report these results. The results for large-scale binary packing instances are shown in Table 3-Table 5, where only the cases with the Gap greater than 10^{-4} or running time within the time limit are displayed. Overall, we see that increasing the number of SOC constraints (i.e., N) can improve the Gap effectively but also requires a longer running time. In this experiment, two of our best methods (i.e., Example 3 Best Scale and Section 3.2 Shift) on average are 85% shorter than MOSEK, with Gap being less than 10^{-4}.

The results for the proposed SOC approximation methods are displayed in Tables 3 and 4. For the binary packing MIECP, Example 1, Section 3.1, Section 3.2, and Section 3.1 Shift methods cannot solve all the cases with the Gap being no larger than 10^{-4} within the time limit, while all the other methods are consistently better than MOSEK. Particularly, it is worthy of mentioning that Example 1 with $N = 20$ has numerical issues, and we are not able to improve the solution quality by increasing N. On average, the running time of Example 2 and Example 3 can be 60%, 69% shorter than that of MOSEK, respectively. Both Example 3 with Best Scale and Section 3.2 Shift method apparently are the best among all the methods, since their average running time is only 13% of that of MOSEK. Particularly, Example 3 with Best Scale only needs one SOC constraint, and Section 3.2 Shift method needs two SOC constraints to approximate each exponential conic constraint. It is worthy of emphasizing that we find the approximate solution for Example 3 with Best Scale and Section 3.2 Shift by executing Example 3 with the approximate solution equal to 2^{-4} and 2-second time limit. We also see that Section 3.1 and Section 3.2 methods run into numerical issues, while the shifting method improves them significantly.

The results for the proposed polyhedral approximation methods are shown in Table 5. The two polyhedral approximation methods, Cutting Plane and Branch and Cut, perform quite well. Specifically, the average running time of the Cutting Plane method is around 21% of that of MOSEK, and using a delayed cut generation procedure (i.e., Branch and Cut method) has an average running time of around 18% of that of MOSEK.

Experiment 2- Mixed-Integer Packing MIECP: For this experiment, we set $b_i = 2n$ for each $i \in [m], n = 200$, $p \in \{10, 20, \ldots, 100\}$ and let half of the variables be continuous, and the remaining half be binary (i.e., $t = n/2$). The results for large-scale mixed-integer packing cases can be found in Table 6-Table 8, where only the cases with the Gap greater than 10^{-4} or running time within the time limit are displayed. In this experiment, our best method (i.e., Example 3 Best Scale) on average is more than 90% shorter than MOSEK, with Gap being less than 10^{-4}.

Table 3: Numerical Results of Section 2 Methods for Solving the Binary Packing MIECP

$n = 100$	MOSEK		Example 1[i]			Example 2[ii]			Example 3[iii]			Example 3 Best Scale[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	2.63	—	5.34	2.02	—	0.78	0.30	—	2.15	0.81	—	1.71	0.65
10	—	3.63	—	4.85	1.34	—	1.64	0.45	—	1.69	0.47	—	1.65	0.45
15	—	102.83	—	56.03	0.54	—	22.89	0.22	—	15.91	0.15	—	6.54	0.06
20	—	35.12	1.1e-4	78.66	2.24	—	17.23	0.49	—	6.90	0.20	—	5.59	0.16
25	—	111.18	—	512.36	4.61	—	52.48	0.47	—	70.73	0.64	—	22.21	0.20
30	—	742.96	—	1989.90	2.68	—	241.59	0.33	—	188.23	0.25	—	59.96	0.08
35	—	113.84	—	113.64	1.00	—	23.44	0.21	—	15.87	0.14	—	8.04	0.07
40	—	240.44	—	535.56	2.23	—	154.69	0.64	—	90.45	0.38	—	23.77	0.10
45	—	306.41	—	436.13	1.42	—	143.24	0.47	—	68.50	0.22	—	16.73	0.05
50	—	2524.17	8.3e-3	*	*	—	1802.24	0.71	—	940.52	0.37	—	198.66	0.08
Geo Mean					1.72			0.40			0.31			0.13

[i] $N = 20, a = 1$;   [ii] $N = 6, s = 4$;   [iii] $N = 4$ and the approximation solution is 2^{-4};
[iv] $N = 1$ and in the Best Scale procedure, we run Example 3 with $(N = 1, AS = 2^{-4}, TL = 2)$.

Table 4: Numerical Results of Section 3 Methods for Solving the Binary Packing MIECP

$n = 100$	MOSEK		Section 3.1[i]			Section 3.2[ii]			Section 3.1 Shift[iii]			Section 3.2 Shift[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	2.63	1.4e-3	2.80	1.06	2.2e-3	2.82	1.07	—	1.29	0.49	—	2.38	0.91
10	—	3.63	1.5e-3	4.06	1.12	1.6e-3	1.71	0.47	—	1.05	0.29	—	0.81	0.22
15	—	102.83	6.9e-3	5.26	0.05	1.6e-3	10.06	0.10	—	19.50	0.19	—	8.84	0.09
20	—	35.12	1.5e-3	5.79	0.16	2.0e-3	9.36	0.27	—	11.32	0.32	—	5.20	0.15
25	—	111.18	3.8e-3	22.94	0.21	1.7e-3	39.80	0.36	—	769.74	6.92	—	17.37	0.16
30	—	742.96	5.2e-3	64.60	0.09	1.9e-3	125.19	0.17	6.5e-3	*	*	—	82.70	0.11
35	—	113.84	1.5e-3	22.60	0.20	2.2e-3	9.88	0.09	—	440.48	3.87	—	9.60	0.08
40	—	240.44	2.8e-3	85.51	0.36	2.2e-3	961.87	4.00	—	1423.26	5.92	—	17.21	0.07
45	—	306.41	1.0e-2	13.43	0.04	2.2e-3	2213.90	7.23	1.1e-4	1937.42	6.32	—	20.51	0.07
50	—	2524.17	1.2e-2	258.69	0.10	1.8e-2	*	*	1.7e-2	*	*	—	271.84	0.11
Geo Mean					0.19			0.50			1.31			0.13

[i] $N = 15$;   [ii] $N = 6, s = 1$;   [iii] $N = 7$ and using the same approximate solution as Example 3 Best Scale;
[iv] $N = 0, s = 1$ and using the same approximate solution as Example 3 Best Scale.

Table 5: Numerical Results of Section 5 Methods for Solving the Binary Packing MIECP

$n = 100$	MOSEK		Cutting Plane			Branch and Cut		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	2.63	—	1.36	0.52	—	0.96	0.37
10	—	3.63	—	1.23	0.34	—	1.93	0.53
15	—	102.83	—	19.31	0.19	—	12.41	0.12
20	—	35.12	—	7.38	0.21	—	6.06	0.17
25	—	111.18	—	23.91	0.22	—	42.94	0.39
30	—	742.96	—	147.79	0.20	—	116.78	0.16
35	—	113.84	—	12.86	0.11	—	13.33	0.12
40	—	240.44	—	38.89	0.16	—	23.58	0.10
45	—	306.41	—	38.90	0.13	—	18.52	0.06
50	—	2524.17	—	716.95	0.28	—	607.11	0.24
Geo Mean					0.21			0.18

The results for the proposed SOC approximation methods are shown in Tables 6 and 7. For the mixed-integer packing MIECP, both Section 3.1 and Section 3.2 methods cannot solve all the cases with the Gap being no larger than 10^{-4} within the time limit, while all the other methods are consistently better than MOSEK. The average running time of Example 1, Example 2, and Example 3 is 54%, 67%, 78% shorter than that of MOSEK, respectively. Example 3 with Best Scale is the best among all the methods since its average running time is only 9% of that of MOSEK. Particularly, Example 3 with Best Scale only needs one SOC constraint to approximate each exponential conic constraint. It is worthy of emphasizing that we find the approximate solution for Example 3 with Best Scale, denoted as Best Scale procedure, by executing Example 3 with the approximate solution equal to 2^{-4} and 2-second time limit. We also see that both Section 3.1 and Section 3.2 methods run into numerical issues, while the shifting method improves them significantly.

Table 6: Numerical Results of Section 2 Methods for Solving the Mixed-Integer Packing MIECP

$n = 200$	MOSEK		Example 1[i]			Example 2[ii]			Example 3[iii]			Example 3 Best Scale[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
10	—	2.80	—	2.41	0.86	—	1.98	0.71	—	0.71	0.25	—	0.87	0.31
20	—	5.07	—	1.30	0.26	—	1.19	0.23	—	0.86	0.17	—	1.18	0.23
30	—	66.96	—	19.19	0.29	—	11.51	0.17	—	11.02	0.16	—	5.22	0.08
40	—	158.62	—	60.74	0.38	—	34.48	0.22	—	31.83	0.20	—	10.30	0.06
50	—	119.30	—	57.94	0.49	—	30.80	0.26	—	19.87	0.17	—	6.59	0.06
60	—	945.54	—	209.07	0.22	—	148.00	0.16	—	98.70	0.10	—	32.98	0.03
70	—	1774.46	—	583.00	0.33	—	561.01	0.32	—	292.50	0.16	—	88.63	0.05
80	—	224.86	—	81.89	0.36	—	76.54	0.34	—	67.31	0.30	—	19.19	0.09
90	—	1636.50	—	778.97	0.48	—	499.44	0.31	—	363.26	0.22	—	83.93	0.05
100	—	393.98	—	1062.28	2.70	—	702.11	1.78	—	401.33	1.02	—	86.15	0.22
Geo Mean					0.46			0.33			0.22			0.09

[i] $N = 8, a = 1$;   [ii] $N = 6, s = 4$;   [iii] $N = 3$ and the approximation solution is 2^{-4};
[iv] $N = 1$ and in the Best Scale procedure, we run Example 3 with ($N = 1, AS = 2^{-4}, TL = 2$).

Table 7: Numerical Results of Section 3 Methods for Solving the Mixed-Integer Packing MIECP

$n = 200$	MOSEK		Section 3.1[i]			Section 3.2[ii]			Section 3.1 Shift[iii]			Section 3.2 Shift[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
10	—	2.80	2.4e-3	2.22	0.79	1.2e-4	1.88	0.67	—	2.48	0.89	—	0.97	0.35
20	—	5.07	2.3e-3	9.69	1.91	—	2.26	0.44	—	4.55	0.90	—	1.46	0.29
30	—	66.96	2.2e-3	31.34	0.47	—	8.61	0.13	—	13.65	0.20	—	7.03	0.10
40	—	158.62	2.2e-3	53.16	0.34	—	21.24	0.13	—	29.98	0.19	—	15.22	0.10
50	—	119.30	2.1e-3	59.63	0.50	—	15.00	0.13	—	24.24	0.20	—	16.01	0.13
60	—	945.54	2.2e-3	459.42	0.49	2.3e-4	48.02	0.05	—	128.16	0.14	—	43.6	0.05
70	—	1774.46	2.2e-3	489.68	0.28	2.6e-4	99.34	0.06	—	297.84	0.17	—	81.96	0.05
80	—	224.86	2.1e-3	88.69	0.39	—	37.06	0.16	—	33.04	0.15	—	17.51	0.08
90	—	1636.50	2.1e-3	969.90	0.59	1.2e-4	136.14	0.08	—	2215.28	1.35	—	111.63	0.07
100	—	393.98	2.1e-3	963.85	2.45	1.4e-4	157.30	0.40	—	367.79	0.93	—	115.81	0.29
Geo Mean					0.63			0.16			0.35			0.12

[i] $N = 20$;   [ii] $N = 6, s = 1$;   [iii] $N = 5$ and using the same approximate solution as Example 3 Best Scale;   [iv] $N = 0, s = 1$ and using the same approximate solution as Example 3 Best Scale.

The results for the proposed polyhedral approximation methods are shown in Table 8. The polyhedral approximation method, i.e., Branch and Cut, performs well. Specifically, the average running time of the Branch and Cut method is around 27% of that of MOSEK.

Table 8: Numerical Results of Section 5 Methods for Solving the Mixed-Integer Packing MIECP

$n = 200$	MOSEK		Branch and Cut		
p	Gap	Time (s)	Gap	Time (s)	Ratio
10	—	2.80	—	2.00	0.71
20	—	5.07	—	3.02	0.60
30	—	66.96	—	15.87	0.24
40	—	158.62	—	29.90	0.19
50	—	119.30	—	16.97	0.14
60	—	945.54	—	159.46	0.17
70	—	1774.46	—	238.71	0.13
80	—	224.86	—	45.94	0.20
90	—	1636.50	—	361.91	0.22
100	—	393.98	—	323.73	0.82
Geo Mean					0.27

6.2 Covering MIECP

In this subsection, we consider the following mixed-integer convex covering problem

$$\min_{\boldsymbol{x} \in X} \left\{ \sum_{\ell \in [p]} \left(\sum_{j \in [n]} c_{\ell j} x_j \right) \log \left(\sum_{j \in [n]} c_{\ell j} x_j \right) : \sum_{j \in [n]} a_{ij} x_j \geq b_i, \forall i \in [m] \right\}, \tag{22}$$

where $X = \{0,1\}^t \times [0,1]^{n-t}$. Note that the problem (22) can be formulated as the following covering MIECP

$$\min_{\boldsymbol{x} \in X} \left\{ \sum_{\ell \in [p]} v_\ell : \left(1, \sum_{j \in [n]} c_{\ell j} x_j, -v_\ell\right) \in K_{\exp}(0), \forall \ell \in [p], \sum_{j \in [n]} a_{ij} x_j \geq b_i, \forall i \in [m] \right\}. \tag{23}$$

Similar to the packing MIECP, we conduct two experiments to test the proposed SOC and polyhedral approximations for solving (23), where we consider pure binary ($t = n$) and mixed-integer ($t \in [n-1]$) experiments and compare our results with MOSEK. In our testing instances, we set $m = 100$ and suppose $a_{ij} \sim \text{int}(0,9), b_i = 2n, c_{\ell j} \sim \text{int}(0,9)/n$ for all $i \in [m], j \in [n], \ell \in [p]$. The time limit is set to be 3600 seconds for both experiments.

Experiment 3- Binary Covering MIECP: For the binary experiment, we solve the small-scale instances with $n = 30, p \in \{5, 10, \ldots, 50\}$ to do sensitive analyses of parameters and learn the patterns and then solve the large-scale instances with $n = 50, p \in \{5, 10, \ldots, 50\}$. Due to the page limit, we do not report these results. The results for large-scale binary covering instances can be found in Table 9-Table 11, where only the cases with the Gap greater than 10^{-4} or running time within the time limit are displayed. In this experiment, our best method (i.e., Branch and Cut), on average, is at least 90% shorter than MOSEK, with Gap being less than 10^{-4}.

The results for the proposed SOC approximation methods are shown in Tables 9 and 10. It is seen that the Section 3.1 method cannot solve all the cases with the Gap being no larger than 10^{-4} within the time limit, while all the other methods are consistently better than MOSEK. On average, the running time of Example 1, Example 2, and Example 3 can be $67\%, 79\%, 80\%$ shorter than that of MOSEK, respectively. It is worthy of mentioning that Example 3 with Best Scale is 88% and Section 3.2 Shift method is 89% shorter than MOSEK. Particularly, both Example 3 with Best Scale and Section 3.2 Shift method only need one SOC constraint to approximate each exponential conic constraint. It is worthy of emphasizing that we find the approximate solution for Example 3 with Best Scale and Section 3.2 Shift by executing Example 3 with the approximate solution equal to 2^{-1} and 2-second time limit. We also see that the Section 3.1 method runs into numerical issues, while the shifting method improves it significantly.

Table 9: Numerical Results of Section 2 Methods for Solving the Binary Covering MIECP

$n = 50$	MOSEK		Example 1[i]			Example 2[ii]			Example 3[iii]			Example 3 Best Scale[iv]	
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	0.89	—	0.24	0.27	—	0.20	0.22	—	0.33	0.37	—	1.86	2.09
10	—	18.30	—	5.42	0.30	—	3.79	0.21	—	2.16	0.12	—	3.51	0.19
15	—	49.04	—	18.60	0.38	—	9.76	0.20	—	7.78	0.16	—	5.83	0.12
20	—	159.79	—	39.52	0.25	—	23.44	0.15	—	20.51	0.13	—	10.91	0.07
25	—	368.94	—	59.29	0.16	—	54.23	0.15	—	30.71	0.08	—	21.49	0.06
30	—	720.02	—	332.12	0.46	—	178.82	0.25	—	156.53	0.22	—	61.50	0.09
35	—	313.22	—	100.84	0.32	—	58.94	0.19	—	77.35	0.25	—	21.71	0.07
40	—	827.92	—	376.44	0.45	—	192.63	0.23	—	334.41	0.40	—	74.76	0.09
45	—	532.81	—	216.65	0.41	—	163.42	0.31	—	139.18	0.26	—	33.33	0.06
50	—	307.16	—	145.28	0.47	—	65.50	0.21	—	58.03	0.19	—	25.15	0.08
Geo Mean					0.33			0.21			0.20			0.12

[i] $N = 4, a = 1$;   [ii] $N = 3, s = 1$;   [iii] $N = 2$ and the approximation solution is 2^{-1};
[iv] $N = 1$ and in the Best Scale procedure, we run Example 3 with 3-tuple ($N = 1, AS = 2^{-1}, TL = 2$).

The results for the proposed polyhedral approximation methods are shown in Table 11. The two polyhedral approximation methods, i.e., Cutting Plane and Branch and Cut, perform very well, and the Branch and Cut method is the best among all the methods. Specifically, the average running time of the Cutting Plane method is around 7% of that of MOSEK, and using delayed cut generation procedure (i.e., Branch and Cut method) has an average running time of around 6% of that of MOSEK.

Experiment 4- Mixed-Integer Covering MIECP: For the mixed-integer experiment, we set $n = 200, p \in \{10, 20, \ldots, 100\}$ and let half of the variables be continuous, and the remaining half be binary (i.e., $t = n/2$). The results for the mixed-integer covering instances can be found in Table 12-Table 14, where only the cases

Table 10: Numerical Results of Section 3 Methods for Solving the Binary Covering MIECP

$n = 50$	MOSEK		Section 3.1[i]			Section 3.2[ii]			Section 3.1 Shift[iii]			Section 3.2 Shift[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	0.89	7.1e-4	0.27	0.31	—	6.75	7.61	—	0.48	0.54	—	0.48	0.55
10	—	18.30	8.6e-4	9.43	0.52	—	5.29	0.29	—	7.59	0.41	—	3.39	0.19
15	—	49.04	9.4e-4	13.32	0.27	—	12.42	0.25	—	12.75	0.26	—	6.68	0.14
20	—	159.79	1.0e-3	32.51	0.20	—	31.19	0.20	—	26.63	0.17	—	8.14	0.05
25	—	368.94	1.0e-3	1167.74	3.17	—	1096.47	2.97	—	44.59	0.12	—	22.54	0.06
30	—	720.02	1.0e-3	2750.53	3.82	—	163.64	0.23	—	126.02	0.18	—	76.89	0.11
35	—	313.22	9.2e-4	1661.78	5.31	—	106.81	0.34	—	61.83	0.20	—	31.21	0.10
40	—	827.92	9.6e-4	2595.93	3.14	—	536.93	0.65	—	164.26	0.20	—	66.03	0.08
45	—	532.81	9.6e-4	1638.89	3.08	—	2047.01	3.84	—	164.99	0.31	—	43.12	0.08
50	—	307.16	9.2e-4	1126.08	3.67	—	1366.49	4.45	—	893.44	2.91	—	37.14	0.12
Geo Mean					1.35			0.88			0.31			0.11

[i] $N = 9$;   [ii] $N = 6, s = 1$;   [iii] $N = 6$ and using the same approximate solution as Example 3 Best Scale;   [iv] $N = 0, s = 1$ and using the same approximate solution as Example 3 Best Scale.

Table 11: Numerical Results of Section 5 Methods for Solving the Binary Covering MIECP

$n = 50$	MOSEK		Cutting Plane			Branch and Cut		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
5	—	0.89	—	0.14	0.16	—	0.37	0.42
10	—	18.30	—	2.96	0.16	—	2.32	0.13
15	—	49.04	—	3.48	0.07	—	4.67	0.10
20	—	159.79	—	7.40	0.05	—	5.76	0.04
25	—	368.94	—	30.03	0.08	—	10.23	0.03
30	—	720.02	—	58.17	0.08	—	27.26	0.04
35	—	313.22	—	12.68	0.04	—	13.28	0.04
40	—	827.92	—	56.95	0.07	—	36.91	0.04
45	—	532.81	—	35.53	0.07	—	22.45	0.04
50	—	307.16	—	9.55	0.03	—	21.22	0.07
Geo Mean					0.07			0.06

with the Gap greater than 10^{-4} or running time within the time limit are displayed. In this experiment, two of our best methods (i.e., Example 3 Best Scale and Section 3.2 Shift) on average are around 89% shorter than MOSEK, with Gap being less than 10^{-4}.

The results for the proposed SOC approximation methods are shown in Tables 12 and 13. For mixed-integer covering MIECP, Section 3.1 and Section 3.2 methods cannot solve all the cases with the Gap being no larger than 10^{-4} within the time limit, while all the other methods are consistently better than MOSEK. On average, the running time of Example 1, Example 2, and Example 3 can be 79%, 82%, 87% shorter than that of MOSEK, respectively. Both Example 3 with Best Scale and Section 3.2 Shift methods are the best among all the methods since their average running time is only 11% of that of MOSEK. Particularly, both Example 3 with Best Scale and Section 3.2 Shift only need one SOC constraint to approximate each exponential conic constraint. It is worthy of emphasizing that we find the approximate solution for Example 3 with Best Scale and Section 3.2 Shift method by executing Example 3 with the approximate solution equal to 2^{-1} and 2-second time limit. We also see that Section 3.1 and Section 3.2 methods run into numerical issues, while the shifting method improves them significantly.

The results for the proposed polyhedral approximation methods are shown in Table 14. The polyhedral approximation method, i.e., Branch and Cut, performs very well. Specifically, the average running time of the Branch and Cut method is around 12% of that of MOSEK.

6.3 Sparse Logistic Regression (SLR)

In this subsection, we consider the following sparse logistic regression (SLR)

$$\min_{\substack{\boldsymbol{\theta} \in \mathbb{R}^d, \\ \boldsymbol{z} \in \{0,1\}^d}} \left\{ \sum_{i \in [n]} \left[-y_i \log(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) - (1 - y_i) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \right] + \lambda \|\boldsymbol{\theta}\|_1 : \begin{matrix} \sum_{j \in [d]} z_j = k, \\ \boldsymbol{\theta} - M_{\theta} \boldsymbol{z} \leq 0, \\ -\boldsymbol{\theta} - M_{\theta} \boldsymbol{z} \leq 0 \end{matrix} \right\}, \quad (24)$$

Table 12: Numerical Results of Section 2 Methods for Solving the Mixed-Integer Covering MIECP

$n = 200$	MOSEK		Example 1[i]			Example 2[ii]			Example 3[iii]			Example 3 Best Scale[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
10	—	3.40	—	1.10	0.32	—	0.79	0.23	—	0.71	0.21	—	1.72	0.50
20	—	8.38	—	1.56	0.19	—	3.64	0.43	—	1.17	0.14	—	2.77	0.33
30	—	74.58	—	10.33	0.14	—	8.85	0.12	—	7.24	0.10	—	6.00	0.08
40	—	130.38	—	20.33	0.16	—	14.47	0.11	—	12.29	0.09	—	8.58	0.07
50	—	198.82	—	26.63	0.13	—	19.20	0.10	—	15.73	0.08	—	11.99	0.06
60	—	79.42	—	19.87	0.25	—	12.40	0.16	—	9.84	0.12	—	7.67	0.10
70	—	113.44	—	21.54	0.19	—	15.18	0.13	—	12.45	0.11	—	9.01	0.08
80	—	1515.42	—	183.89	0.12	—	161.97	0.11	—	105.55	0.07	—	60.74	0.04
90	—	438.67	—	82.62	0.19	—	54.26	0.12	—	62.93	0.14	—	27.16	0.06
100	—	357.46	—	293.19	0.82	—	304.25	0.85	—	191.73	0.54	—	110.71	0.31
Geo Mean					0.21			0.18			0.13			0.11

[i] $N = 3, a = 1$;   [ii] $N = 3, s = 1$;   [iii] $N = 2$ and the approximation solution is 2^{-1};
[iv] $N = 1$ and in the Best Scale procedure, we run Example 3 with 3-tuple ($N = 1, AS = 2^{-1}, TL = 2$).

Table 13: Numerical Results of Section 3 Methods for Solving the Mixed-Integer Covering MIECP

$n = 200$	MOSEK		Section 3.1[i]			Section 3.2[ii]			Section 3.1 Shift[iii]			Section 3.2 Shift[iv]		
p	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
10	—	3.40	—	13.93	4.09	—	1.79	0.52	—	1.98	0.58	—	1.63	0.48
20	—	8.38	—	36.36	4.34	—	4.92	0.59	—	3.12	0.37	—	3.54	0.42
30	—	74.58	—	242.64	3.25	—	12.97	0.17	—	9.41	0.13	—	6.69	0.09
40	—	130.38	4.9e-4	376.26	2.89	—	24.50	0.19	—	11.64	0.09	—	8.54	0.07
50	—	198.82	2.1e-3	400.18	2.01	—	30.46	0.15	—	18.31	0.09	—	12.10	0.06
60	—	79.42	—	204.16	2.57	—	21.50	0.27	—	13.52	0.17	—	7.13	0.09
70	—	113.44	—	305.38	2.69	—	197.95	1.74	—	13.44	0.12	—	8.83	0.08
80	—	1515.42	1.7e-3	*	*	1.5e-3	*	*	—	87.86	0.06	—	51.44	0.03
90	—	438.67	1.5e-4	1825.29	4.16	—	1633.22	3.72	—	47.07	0.11	—	22.49	0.05
100	—	357.46	3.0e-3	*	*	2.0e-3	*	*	—	144.77	0.41	—	85.60	0.24
Geo Mean					3.15			0.48			0.16			0.11

[i] $N = 12$;   [ii] $N = 6, s = 1$;   [iii] $N = 3$ and using the same approximate solution as Example 3 Best
Scale;   [iv] $N = 0, s = 1$ and using the same approximate solution as Example 3 Best Scale.

Table 14: Numerical Results of Section 5 Methods for Solving the Mixed-Integer Covering MIECP

$n = 200$	MOSEK		Branch and Cut		
p	Gap	Time (s)	Gap	Time (s)	Ratio
10	—	3.40	—	1.76	0.52
20	—	8.38	—	1.68	0.20
30	—	74.58	—	7.45	0.10
40	—	130.38	—	17.42	0.13
50	—	198.82	—	13.22	0.07
60	—	79.42	—	5.30	0.07
70	—	113.44	—	8.02	0.07
80	—	1515.42	—	137.53	0.09
90	—	438.67	—	24.57	0.06
100	—	357.46	—	111.92	0.31
Geo Mean					0.12

where there are n data points $\{(\boldsymbol{x}_i, y_i)\}_{i \in [n]} \subseteq \mathbb{R}^d \times \{0, 1\}$, $h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = [1 + \exp(-\boldsymbol{\theta}^\top \boldsymbol{x}_i)]^{-1}$ denotes the sigmoid function, and M_θ is the big-M coefficient. Note that we can choose $M_\theta := n \log(2)/\lambda$. Indeed, observe that the optimal value of SLR (24) must be less than or equal to $n \log(2)$ since $\boldsymbol{\theta} = \boldsymbol{0}$ is a feasible solution with objective value equal to $n \log(2)$. Thus, there exists an optimal solution such that

$$\sum_{i \in [n]} \left[-y_i \log(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) - (1 - y_i) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \right] + \lambda \|\boldsymbol{\theta}\|_1 \leq n \log(2).$$

Since $\sum_{i \in [n]} \left[-y_i \log(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) - (1 - y_i) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \right] \geq 0$, we must have $\lambda \|\boldsymbol{\theta}\|_1 \leq n \log(2)$. Therefore, we can upper bound $\|\boldsymbol{\theta}\|_1$ as $\|\boldsymbol{\theta}\|_1 \leq n \log(2)/\lambda = M_\theta$, i.e., $|\theta_j| \leq M_\theta$ is valid for each $j \in [d]$.

Note that SLR (24) can be formulated as the following MIECP

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{z}\in\{0,1\}^d, \boldsymbol{p}_1, \boldsymbol{p}_2} \left\{ \sum_{i\in[n]} t_i + \lambda\|\boldsymbol{\theta}\|_1 : \begin{array}{c} \sum_{j\in[d]} z_j = k, \boldsymbol{\theta} - M_\theta \boldsymbol{z} \le 0, -\boldsymbol{\theta} - M_\theta \boldsymbol{z} \le 0, \\ p_{i1} + p_{i2} = 1, \forall i \in [n], \\ (p_{i1}, 1, (1 - 2y_i)\boldsymbol{\theta}^\top \boldsymbol{x}_i - t_i) \in K_{\exp}(0), \forall i \in [n], \\ (p_{i2}, 1, -t_i) \in K_{\exp}(0), \forall i \in [n] \end{array} \right\}. \quad (25)$$

In this subsection, we conduct two experiments to test the proposed SOC and polyhedral approximations for solving (25), where we consider moderate-scale and large-scale cases and compare our results with MOSEK. In our testing instances, we set $k = 20, \lambda = 0.01$. Overall, the proposed approximation methods can effectively solve SLR using real datasets and outperform MOSEK.

Experiment 5- Moderate-Scale SLR Cases: In this experiment, we solve SLR with the UCI student mathematics performance dataset [14] using MOSEK, Example 3 with Best Scale, and Branch and Cut methods. These three are identified as the best ones in the previous experiments. In particular, in Example 3 with Best Scale, we use the obtained approximate solution to update the M_θ value; and to avoid numerical issues, when implementing the Branch and Cut algorithm, we first run the gradient descent method to solve the continuous relaxation of SLR and then adding all the gradient inequalities into the root node of the branch and bound tree. This dataset contains 31 variables and a binary experimental class, i.e., the positive class for the grade above the median and the negative class for otherwise. We also use the linear kernel to increase features to $d = 496$, and we consider the number of data points being $n \in \{1020, \dots, 100\}$.

Table 15 summarizes the results for moderate-scale SLR instances, where only the cases with the Gap greater than 10^{-4} or running time within the time limit are displayed. Both MOSEK and Example 3 with Best Scale can solve all the cases with the Gap being less than 10^{-4}. Notably, the polyhedral approximation method, i.e., Branch and Cut with the first-order method, can only solve the cases when $n \le 60$ within the time limit of 3600 seconds, and it takes a much longer time than the other two methods. This may be because the big-M coefficient causes numerical difficulty for the MILP solver. Example 3 with Best Scale is the best among all the methods since its average running time is only 5% of that of MOSEK. Particularly, Example 3 with Best Scale needs four SOC constraints

Table 15: Numerical Results of Moderate-Scale SLR

$d = 496$	MOSEK		Example 3 Best Scale[i]			Branch and Cut		
n	Gap	Time (s)	Gap	Time (s)	Ratio	Gap	Time (s)	Ratio
10	—	14.26	—	0.35	0.02	—	10.26	0.72
20	—	21.63	—	1.04	0.05	—	35.42	1.64
30	—	1.90	—	1.84	0.97	—	60.54	31.88
40	—	338.50	—	1.75	0.01	—	65.66	0.19
50	—	9.86	—	4.88	0.49	—	117.96	11.97
60	—	9.05	—	5.90	0.65	—	98.98	10.94
70	—	791.01	—	8.63	0.01	n/a	n/a	n/a
80	—	954.47	—	4.55	0.00	n/a	n/a	n/a
90	—	306.48	—	17.19	0.06	n/a	n/a	n/a
100	—	407.08	—	12.20	0.03	n/a	n/a	n/a
Geo Mean					0.05			3.14

[i] $N = 4$ and in the Best Scale procedure, we first run Example 3 with 3-tuple $(N = 4, AS = 2^{-2}, TL = 10)$.

to approximate each exponential conic constraint. It is worthy of emphasizing that we find the approximate solution for Example 3 with Best Scale by executing Example 3 with the approximate solution equal to 2^{-2} and a 10-second time limit. In summary, in this experiment, our best method (i.e., Example 3 Best Scale) on average is 95% shorter than MOSEK, with Gap being less than 10^{-4}.

Experiment 6- Large-Scale SLR Cases: In this experiment, we solve the UCI oral toxicity dataset [4] with the time limit of 600 seconds using MOSEK and Example 3 with Best Scale methods since Experiment 5 shows that the Branch and Cut method does not work well for SLR cases. Similarly, in Example 3 with Best Scale, we use the obtained approximate solution to update the M_θ value. This dataset contains $d = 1024$ binary variables and one binary experimental class, i.e., the positive class for very toxic and the negative class for not very toxic. We randomly select 500 data points from each class to form a new dataset with 1000 data points and consider $n \in \{100, 200, \dots, 1000\}$. Our solution time limit is set to be 600 seconds since machine learning problems often require finding a good-quality solution within a short amount of time.

Table 16: Numerical Results of Large-Scale SLR

$d = 1024$	MOSEK	Example 3 Best Scale[i]	
n	obj.val	obj.val	obj.impr (%)
100	0.4675	0.4272	8.61
200	0.5303	0.5127	3.31
300	0.5899	0.5315	9.89
400	0.5575	0.5456	2.13
500	0.5512	0.5466	0.83
600	0.5438	0.5408	0.54
700	0.5625	0.5581	0.79
800	0.6014	0.5593	7.01
900	0.5974	0.5684	4.85
1000	0.5673	0.5624	0.87

[i] $N = 1$ and in the Best Scale procedure, we first run Example 3 with 3-tuple $(N = 2, AS = 2^{-2}, TL = 30)$.

Table 16 summarizes the results for large-scale SLR cases, where the objective value (denoted as "obj.val") by substituting the obtained solution into the original SLR objective function in (24) and the relative gap

(denoted as "obj.impr") between the obj.val of MOSEK and the obj.val of Example 3 with Best Scale are displayed. With the time limit of 600 seconds, both methods cannot be solved to optimality; however, Example 3 with Best Scale outputs better objective values than MOSEK. Particularly, Example 3 with Best Scale only needs one SOC constraint to approximate each exponential conic constraint. It is worthy of mentioning that we find the approximate solution for Example 3 with Best Scale by executing Example 3 with the approximate solution equal to 2^{-2} and a 30-second time limit. The relative gaps between the obj.vals of MOSEK and Example 3 Best Scale for different cases vary from 0.54% to 9.89%. Overall, in this experiment, our best method (i.e., Example 3 Best Scale) consistently outperforms MOSEK on solving large-scale SLR instances by providing better quality solutions.

7 Conclusion

This paper studies the approximation schemes of the mixed-integer exponential conic programs (MIECPs). We generalize and extend the existing second-order conic approximation scheme and propose new scaling and shifting methods. We also prove approximation accuracies and derive lower bounds of approximation results. We study the polyhedral outer approximation of the exponential cones in the original space based on gradient inequalities. Our numerical study shows that the scaling, shifting, and polyhedral outer approximation methods work very well and can consistently outperform MOSEK with 5-20 times speed-ups. We are working on developing valid inequalities for MIECPs by exploring submodularity and studying disjunctive cuts.

References

1. F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.
2. M. ApS. Mosek modeling cookbook, 2021.
3. R. Badenbroek and J. Dahl. An algorithm for nonsymmetric conic optimization inspired by mosek. *Optimization Methods and Software*, pages 1–38, 2021.
4. D. Ballabio, F. Grisoni, V. Consonni, and R. Todeschini. Integrated qsar models to predict acute oral systemic toxicity. *Molecular informatics*, 38(8-9):1800124, 2019.
5. A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. SIAM, 2001.
6. A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.
7. G. Blekherman. Nonnegative polynomials and sums of squares. *Journal of the American Mathematical Society*, 25(3):617–635, 2012.
8. S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi. A tutorial on geometric programming. *Optimization and engineering*, 8(1):67–127, 2007.
9. S. P. Boyd, S.-J. Kim, D. D. Patil, and M. A. Horowitz. Digital circuit optimization via geometric programming. *Operations research*, 53(6):899–932, 2005.
10. L. Chen, L. He, and Y. H. Zhou. An exponential cone programming approach for managing electric vehicle charging. *Available at SSRN 3548028*, 2021.
11. M. Chiang. *Geometric programming for communication systems*. Now Publishers Inc, 2005.
12. M. Chiang, C. W. Tan, D. P. Palomar, D. O'neill, and D. Julian. Power control by geometric programming. *IEEE transactions on wireless communications*, 6(7):2640–2651, 2007.
13. C. Coey, M. Lubin, and J. P. Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation*, pages 1–45, 2020.
14. P. Cortez and A. M. G. Silva. Using data mining to predict secondary school student performance. 2008.
15. J. Dahl and E. D. Andersen. A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization. *Mathematical Programming*, pages 1–30, 2021.

16. A. Domahidi, E. Chu, and S. Boyd. Ecos: An socp solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076. IEEE, 2013.
17. H. Fawzi, J. Saunderson, and P. A. Parrilo. Semidefinite approximations of the matrix logarithm. *Foundations of Computational Mathematics*, 19(2):259–296, 2019.
18. P. Jaillet, G. G. Loke, and M. Sim. Strategic manpower planning under uncertainty. *Available at SSRN 3168168*, 2018.
19. B. Kocuk. Conic reformulations for kullback-leibler divergence constrained distributionally robust optimization and applications. *arXiv preprint arXiv:2007.05966*, 2020.
20. M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1-3):193–228, 1998.
21. A. Nemirovski. Advances in convex optimization: conic programming. In *International Congress of Mathematicians*, volume 1, pages 413–444, 2007.
22. B. O'donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.
23. D. Papp and S. Yıldız. alfonso: Matlab package for nonsymmetric conic optimization. *INFORMS Journal on Computing*, 2021.
24. M. Patriksson. A survey on the continuous nonlinear resource allocation problem. *European Journal of Operational Research*, 185(1):1–46, 2008.
25. S. A. Serrano. *Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone*. Stanford University, 2015.
26. A. Skajaa and Y. Ye. A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. *Mathematical Programming*, 150(2):391–422, 2015.
27. V. L. Somers and I. R. Manchester. Sparse resource allocation for control of spreading processes via convex optimization. *IEEE Control Systems Letters*, 5(2):547–552, 2020.
28. L. N. Trefethen. *Approximation theory and approximation practice*, volume 164. Siam, 2019.
29. L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
30. H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer Science & Business Media, 2012.
31. T. Zhu, J. Xie, and M. Sim. Joint estimation and robustness optimization. *Management Science*, 2021.

Appendix A. Proofs

A.1 Proof of Proposition 3

Proposition 3 *The coefficients in* (15) *can be found recursively as*

$$\alpha_s = 1, \beta_s = 1, \sum_{j \in [0,k]} \frac{\alpha_{s-j}\beta_{s-j}^{2k-2j}}{(2k-2j)!} = 1, \sum_{j \in [0,k]} \frac{\alpha_{s-j}\beta_{s-j}^{2k-2j+1}}{(2k-2j+1)!} = 1, \forall k \in [0, s-1]. \tag{16}$$

Proof: We observe that

$$(\beta_k + y)^{2k} = \sum_{i \in [0,2k]} \binom{2k}{i} \beta_k^i y^{2k-i}, \forall k \in [0, s],$$

which allows us to find the coefficients for each $y^k, k \in [0, 2s]$

$$\widehat{\psi}_{N,2s}(y) = \sum_{j \in [0,s]} \frac{\alpha_j}{(2j)!}(\beta_j + y)^{2j}.$$

For $k \in [0, 2s]$, the coefficient for y^k is

$$\begin{cases} \sum_{i \in [0,s-k/2]} \frac{\alpha_{s-i}\beta_{s-i}^{2s-2i-k}}{(2s-2i-k)!k!} & \text{if } k \text{ is even,} \\ \sum_{i \in [0,s-(k+1)/2]} \frac{\alpha_{s-i}\beta_{s-i}^{2s-2i-k}}{(2s-2i-k)!k!} & \text{if } k \text{ is odd.} \end{cases}$$

In the expression of $\widehat{\psi}_{N,2s}(y) = \sum_{i \in [0,2s]} y^i/i!$, we also know that the coefficient for y^k is $1/k!$ for each $k \in [0, 2s]$. Since

$$\widehat{\psi}_{N,2s}(y) = \sum_{i \in [0,2s]} \frac{y^i}{i!} = \sum_{j \in [0,s]} \frac{\alpha_j}{(2j)!}(\beta_j + y)^{2j},$$

for each $k \in [0, 2s]$, we can obtain

$$\frac{1}{k!} = \begin{cases} \sum_{i \in [0,s-k/2]} \frac{\alpha_{s-i}\beta_{s-i}^{2s-2i-k}}{(2s-2i-k)!k!} & \text{if } k \text{ is even,} \\ \sum_{i \in [0,s-(k+1)/2]} \frac{\alpha_{s-i}\beta_{s-i}^{2s-2i-k}}{(2s-2i-k)!k!} & \text{if } k \text{ is odd.} \end{cases}$$

Solving these equations, we arrive at the conclusion. □

Appendix B. Small-Scale Binary Packing MIECP Results

Figure 5 shows the Gap and running time for Example 1 with $N \in \{10, 15, 20\}$ and $a \in \{0.90, 0.95, \ldots, 1.10\}$. As shown in Figures 5a-5c, for a given a, the Gap can be improved in general by increasing N. We see that the gap is greater than 10^{-4} for all a when $N = 10$. When $N = 15$, using Example 1 with $a \in \{1.00, 1.05, 1.10\}$ can solve several cases with the gap being no larger than 10^{-4}. When $N = 20$, Example 1 with $a \in \{1.00, 1.05, 1.10\}$ can solve all the cases with the Gap being within 10^{-4}. We notice that when N increases, the improvement of Gap when $a \in \{0.90, 0.95\}$ is not significant compared to the other a values. In particular, since Gurobi uses 10^{-4} as the default Gap, the improvement due to increasing N might not be evident; see $a = 1.05, 1.10$ in Figures 5b and 5c, for instance. As shown in Figures 5d-5f, Gurobi spends a longer time on solving the problem with larger p or N values in general, since a larger p or N value means more SOC constraints. Given the same setting of p and N, the running time using Example 1 for different a values is nearly the same. On average, Gurobi spends around 0.5, 1.0, 1.5 seconds on solving packing MIECP (21) using Example 1 with $N = 10, 15, 20$, respectively. Hence, overall, we see that choosing $a = 1$ and $N = 20$ might be desirable when solving larger instances.

(a) Gap for $N = 10$　　　　　　　　　(b) Gap for $N = 15$　　　　　　　　　(c) Gap for $N = 20$

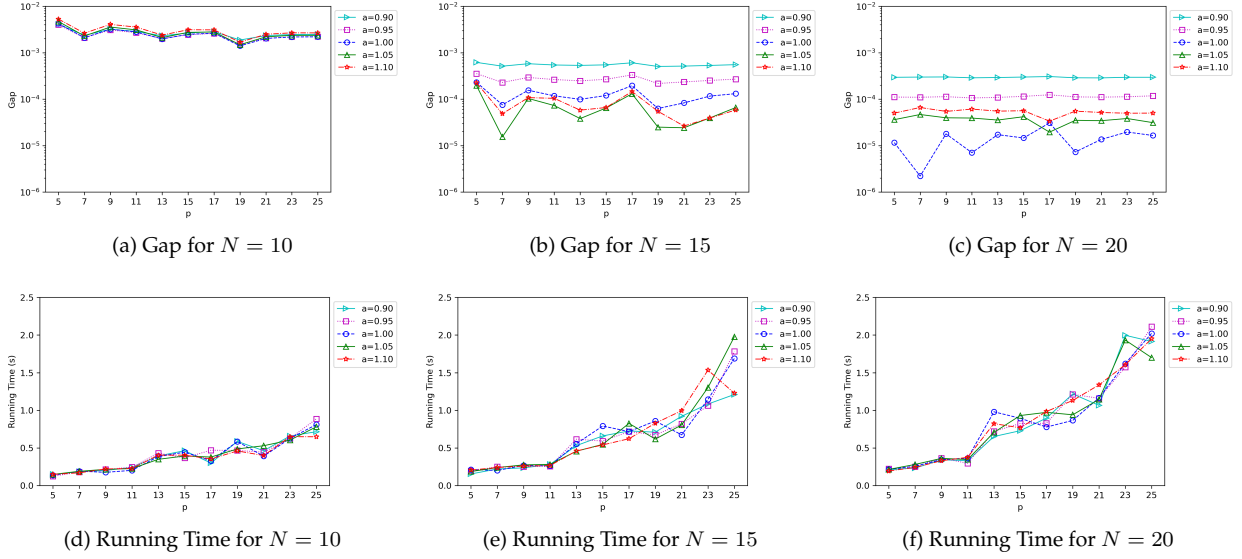(d) Running Time for $N = 10$　　　　(e) Running Time for $N = 15$　　　　(f) Running Time for $N = 20$

Fig. 5: Numerical Illustration of Example 1: Small-Scale Binary Packing MIECP

Figure 6 shows the Gap and running time for Example 2 with $N \in \{1, 2, 3\}$ and $s \in \{1, 2, 3, 4, 5\}$. As shown in Figures 6a-6c, for each s, the Gap can be improved in general by increasing N. We see that the gap is greater than 10^{-4} for all the s when $N = 1$. When $N = 2$, Example 2 with $s \in \{4, 5\}$ can solve all the cases with Gap being no larger than 10^{-4}. When $N = 3$, Example 2 with $s \in \{3, 4, 5\}$ can solve all the cases with Gap within 10^{-4}. With the same N, Example 2 using a larger s value can solve the cases with a better Gap. However, due to the default setting of Gurobi, the improvement due to increasing N or s might be unpredictable; see $s = 4, 5$ in Figures 6b, 6c for instance. As shown in Figures 6d-6f, Gurobi spends a longer time solving the instances with larger p or N values in general. For instance, with fixed values p and N, using Example 2 with larger s values tends to spend more time on solving the case. The running time of Example 2 with a larger s grows as p increases for the same N. In particular, we observe that using Example 2 with both settings $(N, s) = (2, 4)$ and $(N, s) = (3, 3)$ can solve all the cases within 10^{-4} Gap and their running time is similar. Overall, Example 2 spends a similar amount of time on solving the cases to a certain level of Gap regardless of the changes of N and s. On average, Example 2 spends around $0.2, 0.3, 0.4$ seconds on solving packing MIECP (21) with $N = 1, 2, 3$, respectively. Hence, we recommend that choosing $s = 4$ might be desirable when solving larger instances.

Figure 7 shows the Gap and running time for Example 3 with $N \in \{3, 5, 7\}$ and the approximate solution being selected from the list $\{2^{-6}, 2^{-4}, 2^{-2}, 2^0, \text{Best Scale}\}$. Here, "Best Scale" means that we first solve each case using Example 3 with $N = 1$ and the approximate solution equal to 2^{-4} with a 2-second time limit to find a feasible solution, and then use the found solution to construct the approximate solution for each exponential conic constraint. As shown in Figures 7a-7c, for each approximate solution, the Gap can be improved in general by increasing N. Only Best Scale can solve all the cases with Gap being no larger than 10^{-4} when $N = 3$. When $N = 5$, Example 3 with the approximate solution being selected from $\{2^{-4}, \text{Best Scale}\}$ can solve all the cases with Gap being less than 10^{-4}. When $N = 7$, Example 3 with the approximate solution being selected from $\{2^{-6}, 2^{-4}, \text{Best Scale}\}$ can solve all the cases with Gap within 10^{-4}. We notice that when increasing N, the improvement of Gap for the approximate solution being selected from $\{2^{-2}, 2^0\}$ are not significant compared to other approximate solutions. Given the same N, Example 3 with the approximate solution equal to 2^{-4} can solve the case with a better Gap compared to other approximate solutions except for Best Scale. Remarkably, the Best Scale can solve all the cases with Gap being no larger than 10^{-4} for $N \in \{3, 5, 7\}$ and its gap is around 10^{-7}, which outperforms all the other methods

(a) Gap for $N = 1$ (b) Gap for $N = 2$ (c) Gap for $N = 3$

(d) Running Time for $N = 1$ (e) Running Time for $N = 2$ (f) Running Time for $N = 3$
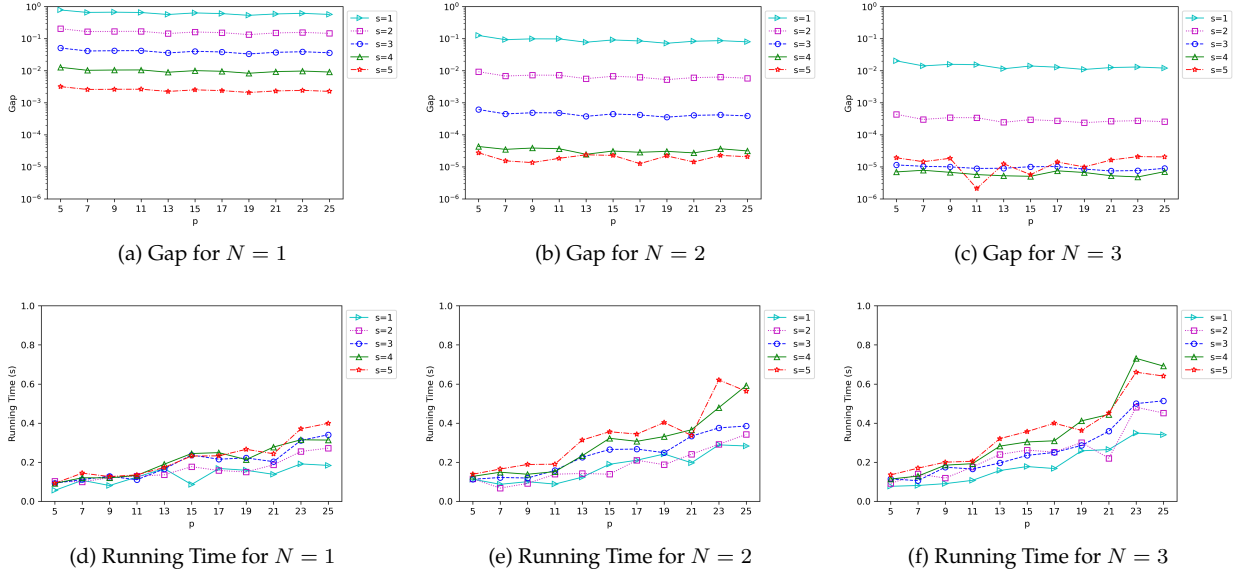
Fig. 6: Numerical Illustration of Example 2: Small-Scale Binary Packing MIECP

based on Example 3 method. In particular, due to the default setting of Gurobi, the improvement due to increasing N might not be obvious; see the cases with the approximate solution equal to 2^{-4}, Best Scale in Figures 7b, 7c for instance. As shown in Figures 7d-7f, Gurobi spends a longer time solving the cases with larger p or N values in general. Given the same setting of p and N, the Best Scale method takes a longer time than other methods based on Example 3 since it needs to solve an additional model to generate a feasible solution. The running time for other methods is similar. However, the Best Scale method still works the best since it requires a much smaller number of SOC constraints to achieve the desired accuracy. Therefore, we suggest using $N = 1$ or 2 for the best case and slightly larger N if we predetermine the approximate solution to be equal to 2^{-4}.

Figure 8 illustrates the Gap and running time for SOC approximations in Section 3 with $N \in \{1, 3, 5\}$. We use "Section 3.1 Shift" and "Section 3.2 Shift" to denote the shifting methods based on the results in Section 3.1 and Section 3.2, respectively. To find a proper approximation solution, we first solve each case using Example 3 with $N = 1$ and the approximate solution 2^{-4} with 2-second time limit to find a feasible solution, and then use this solution to construct the approximate term for each exponential conic constraint. As shown in Figures 8a-8c, the Gap for each method can be improved in general by increasing N, i.e., increasing the number of SOC constraints. However, only can the Section 3.2 Shift method solve all the cases with Gap being less than 10^{-4} when $N = 1$ and its Gap is around 10^{-7}. When $N \in \{3, 5\}$, both Section 3.1 Shift and Section 3.2 Shift methods can solve all the cases with Gap being less than 10^{-4}. Both Section 3.1 and Section 3.2 methods have the Gap being greater than 10^{-4} for all $N \in \{1, 3, 5\}$, which is probably due to the numerical issues. We see that the shifting method can remarkably improve the Section 3.1 and Section 3.2 ones. We also observe that when increasing N, the Gap for the Section 3.2 method decreases more compared to the Section 3.1 method. The Section 3.2 method can solve the cases with a better Gap than that of the Section 3.1 method when $N \in \{3, 5\}$. As shown in Figures 8d-8f, Gurobi spends a longer time solving the cases with larger p or N values in general. Overall, Section 3.2 method spends a longer time on solving the cases than Section 3.1 method, and Section 3.2 Shift method spends a longer time than Section 3.1 Shift method. Although the shifting method increases the running time, we still recommend this method since it can solve the cases with the desirable Gap using fewer points than those without shifting. Besides, no-shifting approaches (i.e., Section 3.1 and Section 3.2) may run into numerical issues. On average, the Section 3.1 and Section 3.2 methods spend around $0.1, 0.2, 0.3$ seconds, and the Section 3.1 Shift and Section
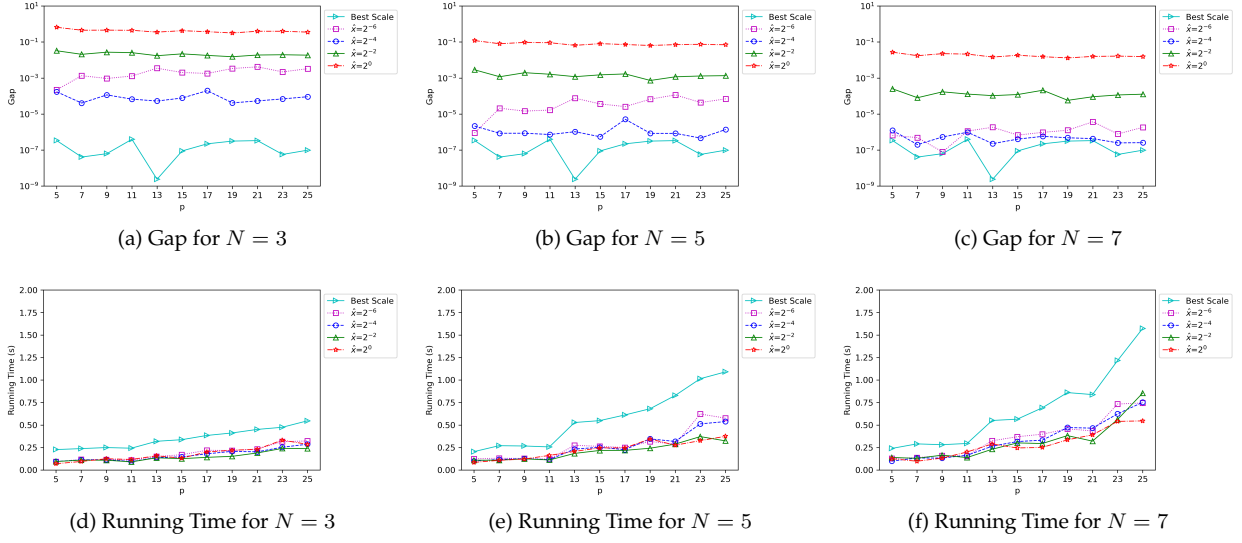
(a) Gap for $N = 3$ (b) Gap for $N = 5$ (c) Gap for $N = 7$

(d) Running Time for $N = 3$ (e) Running Time for $N = 5$ (f) Running Time for $N = 7$

Fig. 7: Numerical Illustration of Example 3: Small-Scale Binary Packing MIECP

3.2 Shift methods spend around $0.3, 0.4, 0.5$ seconds on solving problem (21) with $N = 1, 3, 5$, respectively. Overall, we may only need $N = 1$ or 2 for Section 3.1 Shift and Section 3.2 Shift methods when solving larger instances.
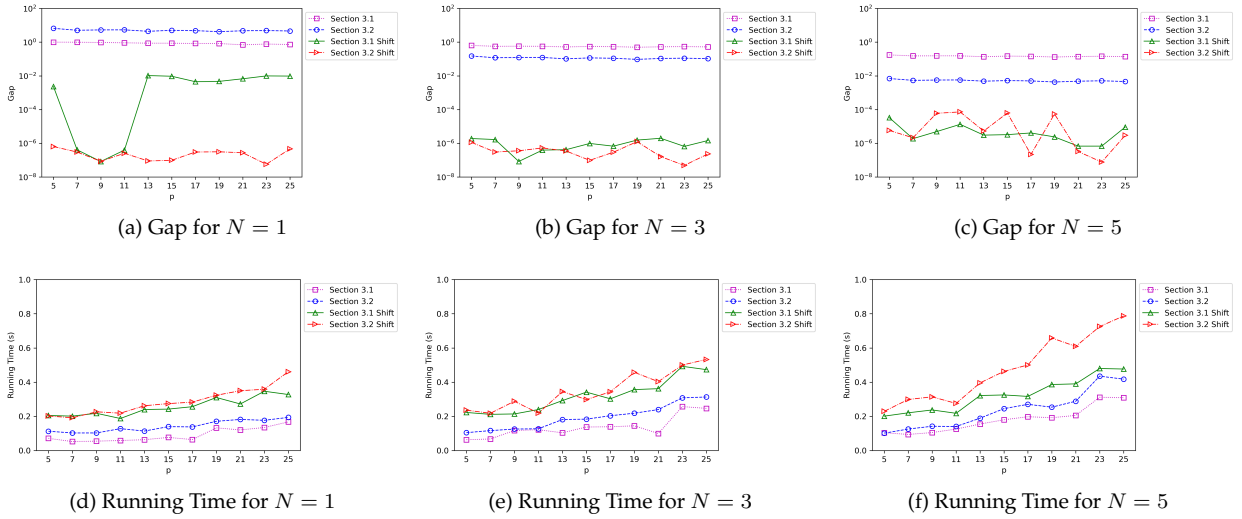

(a) Gap for $N = 1$ (b) Gap for $N = 3$ (c) Gap for $N = 5$

(d) Running Time for $N = 1$ (e) Running Time for $N = 3$ (f) Running Time for $N = 5$

Fig. 8: Numerical Illustration of Section 3 Methods: Small-Scale Binary Packing MIECP

Appendix C. Continuous Packing ECP Results

In this experiment, we test the proposed methods by solving the continuous packing ECP (21), where we consider small-scale and large-scale cases and compare our results with MOSEK. In our testing instances,

we let $t = 0$ and $a_{ij} \sim \text{int}(0,9)$, $b_i = 4n$, $c_{\ell j} \sim -\text{int}(0,9)/n$ for all $i \in [m], j \in [n], \ell \in [p]$, where $\text{int}(p,q)$ denotes a random integer between p and q including p and q. In each case, we compute the relative optimality gap of MOSEK, denoted by "Gap," which is defined as the absolute difference of the ratio of the best objective value from MOSEK over the exact value of the ECP by plugging in the optimal solution found by MOSEK and one. We align our accuracy same as the Gap of MOSEK.

In the small-scale experiment, we consider $n \in \{100, 200, \ldots, 1000\}$, $m = 100$, and we solve 10 cases with $p \in \{10, 20, \ldots, 100\}$ for each n and run Example 3 with $N = 3$ and the approximate solution 2^{-6}. By computing the Gap of MOSEK, we set the accuracy requirement as 10^{-4}. Figure 9a illustrates the average running time for the small-scale continuous packing ECP for each n. Both Example 3 and Cutting Plane methods outperform MOSEK for all the cases, and Example 3 spends a shorter time to solve the cases with larger n compared to Cutting Plane. It is seen that the difference between the average running time of MOSEK and Example 3 increases as n increases.

In the large-scale experiment, we consider $n \in \{200, 400, \ldots, 2000\}$, $m = n$, and we solve 10 cases with $p \in \{200, 400, \ldots, 2000\}$ for each n and run Example 3 with $N = 5$ and the approximate solution 2^{-4}. By computing the Gap of MOSEK, we set the accuracy requirement as 10^{-5}. Figure 9b illustrates the average running time for the large-scale continuous packing ECP for each n. We see that MOSEK is the best among all the methods in this experiment. Meanwhile, Example 3 and Cutting Plane methods work quite well, and these two methods spend a similar amount of time. Compared to MOSEK, the proposed approximation methods take a slightly longer time to solve the continuous problem with larger n. The differences among these three methods are relatively small.

For continuous ECPs, MOSEK performs well, and the proposed methods are comparable to MOSEK. For small-scale problems, Example 3 and Cutting Plane are faster. MOSEK dominates when the problem size is large. While this paper focuses on MIECPs, we highlight that the proposed Cutting Plane method could be considered as an alternative approach for solving small-sized ECPs without parameter tuning.
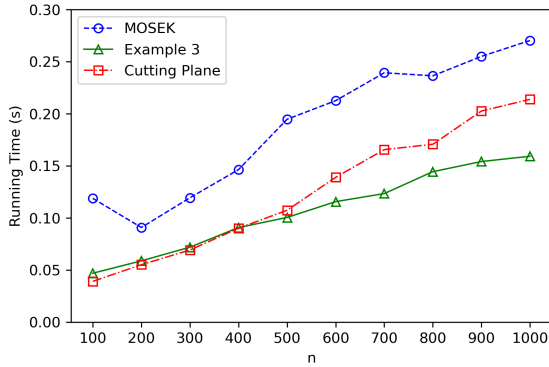
(a) Running Time for Small-Scale Continuous Packing ECP
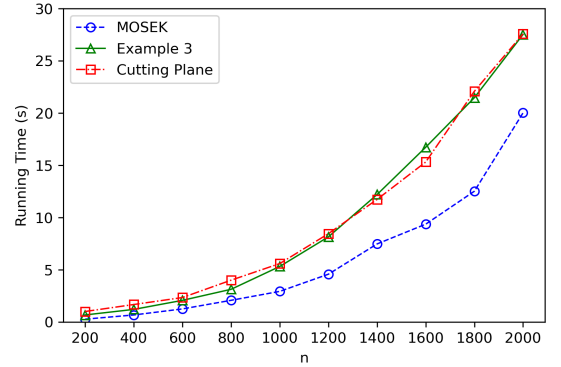
(b) Running Time for Large-Scale Continuous Packing ECP

Fig. 9: Numerical Illustration of MOSEK and Approximation Methods: Continuous Packing ECP