Stochastic graph partitioning: quadratic versus SOCP formulations

Dang Phuong Nguyen, Michel Minoux, Viet Hung Nguyen, Thanh Hai Nguyen, Renaud Sirdey

Noname manuscript No.
(will be inserted by the editor)

Stochastic graph partitioning: Quadratic versus SOCP formulations

Dang Phuong NGUYEN · Michel MINOUX · Viet Hung NGUYEN · Thanh Hai NGUYEN · Renaud SIRDEY

Received: date / Accepted: date

Abstract We consider a variant of the graph partitioning problem involving knapsack constraints with Gaussian random coefficients. In this new variant, under this assumption of probability distribution, the problem can be traditionally formulated as a binary SOCP for which the continuous relaxation is convex. In this paper, we reformulate the problem as a binary quadratic constrained program for which the continuous relaxation is not necessarily convex. We propose several linearization techniques for latter: the classical linearization proposed by Fortet (Trabajos de Estadistica, 1960, Vol. 11) and the linearization proposed by Sherali and Smith (Optimization Letters, 2007, Vol. 1). In addition to the basic implementation of the latter, we propose an improvement which includes, in the computation, constraints coming from the SOCP formulation. Numerical results show that an improvement of Sherali-Smith's linearization outperforms largely the binary SOCP program and the classical linearization when investigated in a branch-and-bound approach.

Keywords Graph Partitioning · Chance Constrained Programming · Central Limit Theorem · Second-Order Cone Programming · Mixed-Integer Linear Programming · Branch-and-Bound

1 Introduction

Graph partitioning problem refers to partitioning the set of nodes of a graph in several disjoint node subsets (or clusters), so that the sum of the weights of the edges whose end-nodes are in different clusters is minimized. A number of variants of the problem have been investigated in the literature depending on constraints imposed on the clusters. We consider here the graph partitioning problem with the knapsack constraints (GPKC) which is defined formally as follows. Given an undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$. The edges in E are weighted by a vector $t \in \mathbb{R}^{|E|}$ and the nodes in V are weighted by a vector $w \in \mathbb{R}^{|V|}$. We want to find a partition of V into $V_1, \ldots, V_k$ such that:

i . For all $1 \leq i \leq k$, $\sum_{v \in V_i} w_v \leq W$ where W is a given constant,

Dang Phuong NGUYEN · Thanh Hai NGUYEN · Renaud SIRDEY
CEA, LIST, Embedded Real Time System Laboratory, Point Courrier 172, 91191 Gif-sur-Yvette, France.
E-mail: dang-phuong.nguyen@cea.fr, thanhhai.nguyen@cea.fr, renaud.sirdey@cea.fr

Michel MINOUX · Viet Hung NGUYEN
LIP6, Pierre and Marie Curie University, 4 Place Jussieu, 75252 Paris Cedex 05, France.
E-mail: michel.minoux@lip6.fr, hung.nguyen@lip6.fr

ii . $\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{\substack{uv \in E \\ u \in V_i \\ v \in V_j}} t_{uv}$ is minimized.

Note that in the above definition, the number k of clusters is not imposed and is a part of the output of the optimization process. The constraints specified in (i) has the structure of the knapsack constraints as reflected in the acronym GPKC. Variants of this problem have already been considered by several authors before, e.g. Sørensen [27], Labbé et al. [20], Bonami et al. [5], Goldschmidt et al. [18]. Besides GPKC, other variants of graph partitioning problem have been investigated involving additional restrictions. Garey et al. [17], Chopra and Rao [7],[8], Ferreira et al. [14] consider the problem of partitioning a graph into at least (resp: no more than), k clusters with no size restriction on the clusters. The cut polytope which is associated with partitioning into two clusters with no size restriction on the clusters is studied by Barahona and Mahjoub [1] and Deza and Laurent [10],[11].

The stochastic version of GPKC referred to as SGPKC, addressed in the present paper considers the case when the node weights are uncertain. These weights follow a multivariate Gaussian distribution with given mean and covariance matrix. Give a probability level $\varepsilon \in [0, 1]$, the knapsack constraints in (i) can be formulated as chance constraints of the form $P(\sum_{v \in V_i} w_v \leq W) \geq 1 - \varepsilon$ for $i = 1, \ldots, k$. This method was introduced in [6] which is one of the standard methods for handling uncertainty in optimization. In the literature, there are some research on stochastic graph partitioning such as [12] and [2], however the uncertainties in their models impact the edges. The present paper seems to be the first study on the stochastic version of the node weighted graph partitioning problem.

Here we investigate SGPKC under the assumption that w follows a multivariate Gaussian distribution. In the case individual chance constraints and with the probability level ε less than 0.5, the chance constraints can then be reformulated as *binary second order cone constraints* (Binary SOCC) [23]. We discuss an application of the above model relative to partitioning of process networks and we explain why the assumption of a Gaussian distribution of the weights is reasonable in this context.

We present a comparison of several alternative techniques for solving SGPKC : -First, we consider the second-order cone formulation for the chance constraint which reduces the SGPKC to a binary second-order cone program (Binary SOCP). The CPLEX solver is used to solve this program. -Second, we consider the quadratic formulation for the chance constraint which handle SGPKC as a binary quadratic constrained program. Several linearization techniques are discussed to transform this binary quadratic program into binary linear program. In particular, we consider the classical linearization technique (Fortet [15]) and the linearization using bilinear forms given by Sherali-Smith [26]. Note that contrary to the former, the latter uses much fewer additional variables. Again the CPLEX solver is used for solving the resulting binary linear programs.

The numerical results obtained show that the solution technique using Sherali-Smith linearization provides better efficiency for SGPKC, than the one using binary SOCP; the latter in turn outperforms the classical linearization technique. This shows that although the quadratic formulation of chance constraint is not convex when relaxed (contrary to the second-order formulation), it provides better efficiency as compared with the second-order cone formulation when the variables are binary and a branch-and-bound procedure has to be applied.

2 Problem statement and Binary SOCP formulation

2.1 Problem formulation

In this section, we propose a 0-1 formulation for SGPKC in which the variables represent the relations between nodes. We use variables x_{ij} for all pairs of nodes (i, j) such that $x_{ij} = 1$ if i and j

are in the same cluster, and 0 otherwise. We denote $E_n := \{(i,j) : i \in V, j \in V, i < j\}$ the set of all ordered pairs of nodes and by $\mathcal{T} := \{(i,j,k) : (i,j),(i,k),(j,k) \in E_n\}$ the set of all ordered triples.

In the deterministic case, the version of graph partitioning problem that we study includes all the the knapsack constraints that express the fact that the total node weight of the cluster containing u should not exceed W. In this paper, we consider the non-deterministic case where the node weights w are random variables. To handle this problem using chance constrained programming, the knapsack constraints are reformulated as chance constraints $P(\sum_{v\in V} x_{vu} w_v \leq W) \geq 1-\varepsilon$ for the cluster containing the node u and with the probability level ε. Hereafter it will be assumed that the probability distribution of node weights is a multivariate Gaussian law with given means $(\bar{w}_i)_{1\leq i\leq n}$ and covariance matrix $(\sigma_{ij})_{1\leq i,j\leq n}$. We therefore reformulate the chance constraints by the Binary SOCCs as follows. For all clusters $i = 1,\ldots,n$,

$$\sum_{u=1}^{n} x_{ui}\bar{w}_u + \gamma\sqrt{\sum_{u=1}^{n}\sigma_{uu}x_{ui}^2 + 2\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}\sigma_{uv}x_{ui}x_{vi}} \leq W \tag{1}$$

where $\gamma = \mathcal{F}^{-1}(1-\varepsilon)$, $\mathcal{F}$ denoting the cumulative distribution function of $\mathcal{N}(0,1)$ (e.g $\gamma \simeq 1.685$ for $\varepsilon = 0.05$). In the proposed model, the various chance constraints on the various clusters are considered as individual chance constraints.

Then the resulting model for SGPKC is the following Binary SOCP program :

$$\text{(I)}\begin{cases}
\min & \displaystyle\sum_{(i,j)\in E} t_{ij}(1-x_{ij}) & \\
\text{s. t.:} & x_{ij} + x_{ik} \leq 1 + x_{jk}, & \forall (i,j,k) \in \mathcal{T} \\
& x_{ij} + x_{jk} \leq 1 + x_{ik}, & \forall (i,j,k) \in \mathcal{T} \\
& x_{ik} + x_{jk} \leq 1 + x_{ij}, & \forall (i,j,k) \in \mathcal{T} \\
& \displaystyle\sum_{u=1}^{n} x_{ui}\bar{w}_u + \gamma\sqrt{\sum_{u=1}^{n}\sigma_{uu}x_{ui}^2 + 2\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}\sigma_{uv}x_{ui}x_{vi}} \leq W & i = 1,\ldots,n \\
& x_{uu} = 1 & u = 1,\ldots,n \\
& x_{ij} \in \{0,1\} & (i,j) \in E_n
\end{cases}$$

where the triangle constraints guarantee the consistency of the partitions, i.e. if the nodes i,j belong to the same cluster and so do the nodes i,k then j and k belong to the same cluster. The number of constraints of (I) is clearly $O(n^3)$, and thus polynomial in terms of n. Hence, it represent a compact 0-1 formulation for the SGPKC. However, [16] shows that the large number of triangle constraints may lead to a huge computation time for the formulation (I) . The possibility of reducing the number of triangle constraints for a sparse graph where $|E| = m \ll \frac{n(n-1)}{2} = |K_n|$ has been investigated in [24] where it is shown that only part of the triangle inequalities are needed. More precisely, let $\mathcal{T}' = \{(i,j,k) : i \neq j \neq k \in V$ and at least one of the edges ij, ik and $jk \in E\}$. Then an equivalent *reduced integer program* formulation for (I) is:

$$
(\text{Bi}-\text{SOCP})\begin{cases}
\min & \displaystyle\sum_{(i,j)\in E} t_{ij}(1-x_{ij}) & \\
\text{s. t.:} & x_{ij}+x_{ik}\leq 1+x_{jk}, & \forall (i,j,k)\in \mathcal{T}^{'}\\
& x_{ij}+x_{jk}\leq 1+x_{ik}, & \forall (i,j,k)\in \mathcal{T}^{'}\\
& x_{ik}+x_{jk}\leq 1+x_{ij}, & \forall (i,j,k)\in \mathcal{T}^{'}\\
& \displaystyle\sum_{u=1}^{n} x_{ui}\bar{w}_u+\gamma\sqrt{\sum_{u=1}^{n}\sigma_{uu}x_{ui}^2+2\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}\sigma_{uv}x_{ui}x_{vi}}\leq W & i=1,\ldots,n\\
& x_{uu}=1 & u=1,\ldots,n\\
& x_{ij}\in\{0,1\} & (i,j)\in E_n
\end{cases}
$$

For a formal proof of the equivalence between (Bi-SOCP) and (I) refer to [24]. It is clear that $|\mathcal{T}^{'}|\leq m(n-2)$ thus the number of triangle constraints in (Bi-SOCP) is $O(mn)$ instead of $O(n^3)$.

We can see that the continuous relaxation of (Bi-SOCP) is a second-order cone program thus it can be solved using SOCP. SOCP has shown its effectiveness in solving nonlinear convex problem that include linear and (convex) quadratic programs as special cases. Several efficient primal-dual interior-point for SOCP have been developed in the last few years which share many of the features of primal-dual interior-point methods for linear programming (LP). However, the algorithmic efficiency of SOCP solvers when embedded into a tree search branch-and-bound to handle Bi-SOCP problem partly remains an open research problem.

2.2 Problem of partitioning process networks and uncertainty of processing time

As a typical example of application of (SGPKC), we can mention a problem arising in the field of compilation for real-time embedded systems, the task allocation problem in multi-core structures. The goal is to find a feasible placement of all tasks to processors while respecting their computing capacity and minimizing the total volume of inter-processor communication. The nodes and edges are weighted by a positive real number could represent the execution time of the task or the volume of data exchanged between tasks. In a partition of the nodes of G, each cluster contains tasks to place in the same processor and it is subject to the type of knapsack constraints that limits the total duration of the tasks in the cluster by a constant. Known for the one-dimensional and deterministic case as the Node Capacitated Graph Partitioning problem [14], the stochastic version of the problem does not seem to have been investigated so far except from a non-parametric and approximate resolution view point [28].

In this problem, one of the main sources of uncertainties lies in the intrinsic indeterminism of execution times for computing kernels of intermediate granularity. This indeterminacy is due firstly to certain characteristics of processor architectures (presence of memories of arbitrators access caches, etc.) but also, inherently, to the presence of conditional branch structures and dependent loops input data. The distributions of processing times are often complex, sometimes giving use to multimodal distributions (due to the presence of data dependent control). However, in spite of the fact that the Gaussian assumption on the random variables is not verified, the use of multivariate Gaussian approximation is still reasonable, based on the extended central limit theorem. Indeed if we assume that for a given u, the number of x_{vu} variables equal to 1 (i.e the cardinality of the cluster containing u) is sufficiently large (typically more than 30-40), the chance constraints

$$P(\sum_{v\in V} x_{vu}w_v\leq W)\geq 1-\varepsilon \quad \forall u\in V$$

involve a combination of sufficiently many random variables, which can be approximated as a normal random variable under some conditions that the means and the variances have to satisfy. These conditions were introduced in the Lindeberg-Feller theorem [21][13] and its corollary, the Liapounov's theorem [22]. This condition was studied for sums of N independent random variables $(X_i)_{1\leq i\leq N}$ with means $(m_i)_{1\leq i\leq N}$ and variances $(a_i^2)_{1\leq i\leq N}$. Let $s_n^2=\sum_{i=1}^N a_i^2$ then :

$$\lim_{N\to\infty}\frac{1}{s_n^2}\sum_{i=1}^{N}\mathbb{E}\left[(X_i-m_i)^2\,\mathbb{1}_{\{|X_i-m_i|>\epsilon s_n\}}\right]=0,\ \forall\epsilon>0\Longrightarrow\frac{\sum_{i=1}^N(X_i-m_i)}{s_n}\xrightarrow{\mathbb{P}}\mathcal{N}(0,1)\qquad(2)$$

Lindeberg's condition is sufficient, but not in general necessary. However if $\lim_{N\to\infty}\max_{i=1,\ldots,N}\frac{a_i^2}{s_n^2}\to 0$, this condition is both sufficient and necessary. For dependent random variables, the central limit theorem remains valid under conditions investigated in [9]. We have carried out some systematic experiments showing that, for sum of N multimodal random variables (three modes were considered in our experiments), good approximations of the Gaussian cdf are obtained as soon as N exceeds typically 30 to 40.

3 Quadratically constrained 0-1 programming reformulation and linearization techniques

As an alternative to Binary SOCP, we investigate here a quadratic formulation for the (SGPKC). To achieve this, we only need to replace the SOCCs (1) in (Bi-SOCP) with their equivalent quadratic forms :

$$\begin{cases}-2\displaystyle\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}(\bar{w}_u\bar{w}_v-\gamma^2\sigma_{uv})x_{ui}x_{vi}+\sum_{u=1}^{n}(2W\bar{w}_u+\gamma^2\sigma_{uu}-\bar{w}_u^2)x_{ui}\leq W^2\\ \displaystyle\sum_{u=1}^{n}x_{ui}\bar{w}_u\leq W\end{cases}\qquad(3)$$

Since the quadratic formulation is difficult to handle directly, we will consider reformulations using various linearization techniques. The first technique discussed below is basically the classical linearization technique [15] and the second one is the linearization technique proposed by Sherali and Smith [26].

We first simplify (3) by setting :

$$\begin{cases}q_{uv}=2(\bar{w}_u\bar{w}_v-\gamma^2\sigma_{uv})\\ d_u=2W\bar{w}_u+\gamma^2\sigma_{uu}-\bar{w}_u^2\end{cases}\qquad(4)$$

then the quadratic constraint in (3) reads :

$$\sum_{u=1}^{n}d_ux_{ui}-\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}q_{uv}x_{ui}x_{vi}\leq W^2\qquad(5)$$

3.1 Classical linearization technique

Introducing variable y_{uvi} to represent each product $x_{ui}x_{vi}$ then (5) is replaced with :

$$\begin{cases}\displaystyle\sum_{u=1}^{n}d_ux_{ui}-\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}q_{uv}y_{uvi}\leq W^2\\ \max\{0,x_{ui}+x_{vi}-1\}\leq y_{uvi}\leq\min\{x_{ui},x_{vi}\}\end{cases}\qquad(6)$$

Using the classical linearization technique, SGPKC can be reformulated as follows :

$$
(\mathrm{CL}) \begin{cases}
\min & \displaystyle\sum_{(i,j)\in E} t_{ij}(1-x_{ij}) & \\
\text{s. t.:} & x_{ij}+x_{ik} \leq 1+x_{jk}, & \forall (i,j,k) \in \mathcal{T}^{'} \\
& x_{ij}+x_{jk} \leq 1+x_{ik}, & \forall (i,j,k) \in \mathcal{T}^{'} \\
& x_{ik}+x_{jk} \leq 1+x_{ij}, & \forall (i,j,k) \in \mathcal{T}^{'} \\
& \displaystyle\sum_{u=1}^{n} x_{ui}\bar{w}_u \leq W & i=1,\ldots,n \\
& \displaystyle\sum_{u=1}^{n} d_u x_{ui} - \sum_{u=1}^{n-1}\sum_{v=u+1}^{n} q_{uv} y_{uvi} \leq W^2 & i=1,\ldots,n \\
& y_{uvi} \leq x_{ui} & \forall u,v,i=1,\ldots,n \\
& y_{uvi} \leq x_{vi} & \forall u,v,i=1,\ldots,n \\
& \max\{0, x_{ui}+x_{vi}-1\} \leq y_{uvi} & \forall u,v,i=1,\ldots,n \\
& x_{uu}=1 & u=1,\ldots,n \\
& x_{ij} \in \{0,1\} & (i,j) \in E_n
\end{cases}
$$

It can easily be shown that the constraints $\max\{0, x_{ui}+x_{vi}-1\} \leq y_{uvi}$ is redundant and can be removed. A drawback of the above formulation (referred to as an "extended formulation" because of the introduction of the extra variables y_{uvi}) is the large number of variables and constraints it requires. Note that in our graph partitioning problem, for a complete graph with n vertices the quadratic formulations we study already have $O(n^2)$ variables and the extended formulation $O(n^3)$ variables and also $O(n^3)$ added constraints. This can become rapidly unpractical. In the following subsection, we discuss an alternative linearization technique requiring fewer additional variables and additional constraints.

3.2 Sherali-Smith's linearization technique

This linearization technique has been introduced in [26]. The basic idea underlying this technique is :

- To transform each quadratic form into a bilinear form using $O(n)$ additional variables: applied to the quadratic constraint (5), it consists in introducing variable λ_{ui} to represent each sum $\displaystyle\sum_{v=u+1}^{n} q_{uv}x_{vi}$. If a lower bound λ_{ui}^{min} and an upper bound λ_{ui}^{max} are known for λ_{ui} then the quadratic constraint (5) can be rewritten as:

$$
\begin{cases}
\sum_{u=1}^{n} d_{ui}x_{ui} - \sum_{u=1}^{n-1} x_{ui}\lambda_{ui} \leq W^2 & \\
\sum_{v=u+1}^{n} q_{uv}x_{vi} = \lambda_{ui}, & \forall u=1,\ldots,n-1 \\
\lambda_{ui}^{min} \leq \lambda_{ui} \leq \lambda_{ui}^{max}, & \forall u=1,\ldots,n-1 \\
x \in \{0,1\}^n &
\end{cases}
$$

– Linearizing the various bilinear terms resulting from the above transformation: this is done by introducing a variable z_{ui} to represent each product $x_{ui}\lambda_{ui}$:

$$\begin{cases} \sum_{u=1}^{n} d_{ui}x_{ui} - \sum_{u=1}^{n-1} z_{ui} \leq W^2 & \\ \sum_{v=u+1}^{n} q_{uv}x_{vi} = \lambda_{ui}, & \forall u = 1,\ldots,n-1 \\ \lambda_{ui}^{min} x_{ui} \leq z_{ui} \leq \lambda_{ui}^{max} x_{ui}, & \forall u = 1,\ldots,n-1 \\ \lambda_{ui}^{min}(1-x_{ui}) \leq \lambda_{ui} - z_{ui} \leq \lambda_{ui}^{max}(1-x_{ui}), & \forall u = 1,\ldots,n-1 \\ x \in \{0,1\}^n & \end{cases}$$

– We focus our study in the case the node weights are all positives (i.e. $w_i \geq 0, \forall i \in V$). This case is suitable for most of the applications of SGPKC including the task allocation problem and to the best of our knowledge, there are no known applications of GPKC with negative node weights. We assume also that the variances are small compared to the means, so that $q_{uv} = 2(\bar{w}_u\bar{w}_v - \gamma^2\sigma_{uv}) > 0, \forall 1 \leq u < v \leq n$. Consequently, we have $\lambda_{ui}^{min} = 0$. Let us introduce new variables $h_{ui} = \lambda_{ui} - z_{ui}$ to the model. A brief analysis shows that these variables are nonnegative and act as slack variables in the contraints $\sum_{v=u+1}^{n} q_{uv}x_{vi} = \lambda_{ui}$ when λ_{ui} is replaced by $h_{ui} + z_{ui}$. Hence, we can ignore the variables h_{ui}'s and the constraints $\lambda_{ui}^{min}(1-x_{ui}) \leq \lambda_{ui} - z_{ui} \leq \lambda_{ui}^{max}(1-x_{ui})$ can be removed. In the general case when the node weights may be negative (i.e. $w_i \in \mathbb{R}, \forall i \in V$), these constraints should not be removed. However, even in this case, the number of constraints of the problem do not increase asymptotically.

Applying the various transformations above to constraints (5) in the quadratic formulation of SGPKC, we can reformulate it as follows:

$$(\text{SS}) \begin{cases} \min & \displaystyle\sum_{(i,j)\in E} t_{ij}(1-x_{ij}) & \\ \text{s. t.:} & x_{ij} + x_{ik} \leq 1 + x_{jk}, & \forall (i,j,k) \in \mathcal{T}' \\ & x_{ij} + x_{jk} \leq 1 + x_{ik}, & \forall (i,j,k) \in \mathcal{T}' \\ & x_{ik} + x_{jk} \leq 1 + x_{ij}, & \forall (i,j,k) \in \mathcal{T}' \\ & \displaystyle\sum_{u=1}^{n} x_{ui}\bar{w}_u \leq W & i = 1,\ldots,n \\ & \displaystyle\sum_{u=1}^{n} d_u x_{ui} - \sum_{u=1}^{n-1} z_{ui} \leq W^2 & i = 1,\ldots,n \\ & \displaystyle z_{ui} \leq \sum_{v=u+1}^{n} q_{uv}x_{vi} & i,u = 1,\ldots,n \\ & 0 \leq z_{ui} \leq \lambda_{ui}^{max} x_{ui} & i,u = 1,\ldots,n \\ & x_{uu} = 1 & u = 1,\ldots,n \\ & x_{ij} \in \{0,1\} & (i,j) \in E_n \end{cases}$$

The above formulation (SS) requires fewer variables and constraints when compared with the classical linearization (CL). For instance, in our problem for a graph with n vertices, the number of added variables and the number of added constraints are $O(n^2)$. However as shown in [26], this is at the expense of weaker relaxation as compared with the classical linearization technique.

The last unknown parameters in this formulation are the bounds λ_{ui}^{min} and λ_{ui}^{max} that can be estimated based on the definition of added variables $(z_{ui})_{1\leq i\leq n-1}$. Note that we assumed $q_{uv} \geq 0$, $\forall (u,v)$ therefore $\lambda_{ui}^{min} = 0$. We will consider two estimates for λ_{ui}^{max} in the following, two variants of the Sherali-Smith reformulation will thus be obtained. In the general case when $q_{uv} \in \mathbb{R}$, $\forall (u,v)$, λ_{ui}^{min} are estimated similarly.

Simple bounds on λ_{ui}^{max}

Note that since $q_{uv} \geq 0,\ \forall (u,v)$, we can take

$$\lambda_{ui}^{max} = \sum_{v=u+1}^{n} q_{uv} \tag{7}$$

It is observed that λ_{ui}^{max} does not depend on i, so at most n values have to be computed. Using these bounds for λ_{ui}^{max} in (SS), we obtain a first formulation based on Sherali-Smith's technique.

Improved bounds on λ_{ui}^{max}

The application of of Sherali-Smith technique can be made more efficient if we can obtain a better estimate of the bounds λ_{ui}^{max}. The idea is the fact that we can get stronger bounds of each sum $\sum_{v=u+1}^{n} q_{uv}x_{vi}$ by adding valid inequalities in the process of computing them. In our problem, one of the valid inequalities that can be chosen is the stochastic knapsack constraint (1), whereby the bounds can be estimated, for all $i = 1, \ldots, n$ as :

$$\lambda_{ui}^{max} = \begin{cases} \max & \displaystyle\sum_{v=u+1}^{n} q_{uv}x_v = \max \sum_{v=u+1}^{n} 2(\bar{w}_u\bar{w}_v - \gamma^2\sigma_{uv})x_v & \\ \text{s. t.:} & \displaystyle\sum_{v=1}^{n} \bar{w}_v x_v + \gamma\sqrt{\sum_{v=1}^{n}\sigma_{vv}x_v^2 + 2\sum_{v=1}^{n-1}\sum_{v'=v+1}^{n}\sigma_{vv'}x_v x_{v'}} \leq W & \\ & x_v \in \{0,1\} & \forall v = 1, \ldots, n \end{cases} \tag{8}$$

Again λ_{ui}^{max} does not depend on i so at most n problems of the form (8) have to be solved. Using the improved bounds λ_{ui}^{max} deduced from (8), we obtain the improved Sherali-Smith formulation (ISS). In addition, we chose the continuous relaxation version of (8) to estimate the bounds λ_{ui}^{max} because our experiments shows that using the integer formulation (8) would not lead to a significant improvement in the quality of the bounds.

4 Computational results

In this section, we present computational results for the SGPKC comparing the formulations that were discussed in the section 4. The formulations are compared in both computation times and gaps.

For a given number of vertices n and number of edges m, we generate five instances by picking edges uniformly at random until the number of edges reaches m. The edge weights t and the means of node weights $\bar{w}$ are drawn independently and uniformly from the interval $[1, 1000]$, the covariance matrix σ of node weights is generated as diagonally dominant matrix where each point $\sigma_{ii}\ \forall i = 1, \ldots, n$ in the diagonal is drawn independently and uniformly from the interval $[1, 20\%\bar{w}_i]$. For each instance, we calibrate the upper bounds of knapsack constraints W in order to ensure that the generated instances will be not "easy" to solved. To achieve this, we used METIS [19] to estimate the solution with the number of clusters $k = 4$ (or 6, 8, 12) that we call the initial partition, we then do 1000 perturbations of this partition. The resulting bounds on W were then chosen so that only 10% of these partitions are satisfied. Finally the probability level ε was chosen to be 0.05 (= 5%), a fairly standard value in practice.

All experiments are run on a machine with Intel Core i7-3630QM 2.40GHz processors and 16 GiB of RAM. The solver CPLEX 12.6 is used to solve respectively (Bi-SOCP), (CL), (SS) and (ISS) and to ensure that comparisons will not be biased we switch off CPLEX pre-solve. All computation times are in CPU seconds and the computation times are subject to a time limit of 7200 seconds.

Table 1 Comparison of the various solution techniques. Nopt indicates that the exact optimal solution could not be found within the imposed time limit (7200s). In such cases, the value of the relative residual gap is shown in parenthesis.

Instances		Bi-SOCP		CL		SS		ISS	
types	n,m	CPU	GAP (Nodes)	CPU	GAP (Nodes)	CPU	GAP (Nodes)	CPU	GAP (Nodes)
SP	25, 40	4.3	16.2 (6)	12.6	17.0 (11)	4.4	18.3 (22)	2.2	18.3 (15)
SP	30, 50	30.8	11.4 (17)	100.9	13.1 (142)	27.5	15.6 (366)	15.6	15.6 (254)
SP	35, 60	40.4	10.5 (32)	177.3	15.4 (336)	43.5	15.9 (829)	29.3	15.9 (599)
SP	40, 65	126.3	9.8 (59)	513.4	11.3 (573)	122.4	12.7 (1136)	63.8	12.7 (729)
SP	45, 75	665.2	13.2 (88)	1539	14.6 (612)	595.0	17.2 (1387)	336.5	17.2 (874)
SP	50, 80	862.8	9.1 (117)	2238	10.4 (935)	978.7	13.3 (1843)	517.2	13.3 (1311)
SP	60, 90	3127	10.6 (156)	Nopt (6.8%)	12.3 (1033)	1844	16.5 (2552)	699.3	16.5 (1566)
SP	80, 130	Nopt (2.4%)	12.7 (187)	Nopt (7.3%)	14.8 (1426)	Nopt (2.8%)	17.3 (3136)	5273	17.3 (3629)
PG	30, 47	105.4	13.2 (49)	256.8	14.6 (395)	97.2	15.3 (866)	41.1	15.3 (542)
PG	40, 66	583.4	12.6 (74)	1727	14.1 (528)	556.2	15.0 (1344)	236.5	15.0 (836)
PG	50, 85	2193	11.4 (125)	Nopt (3.7%)	12.8 (1057)	2386	13.8 (2546)	827.7	13.8 (1888)
PG	60, 104	Nopt (0.2%)	13.2 (146)	Nopt (5.6%)	14.4 (1791)	6216	15.8 (4728)	2397	15.8 (3352)
PG	70, 123	Nopt (2.4%)	12.0 (168)	Nopt (7.1%)	12.2 (2032)	Nopt (3.2%)	14.7 (4759)	6163	14.7 (5292)
TG	30, 60	359.2	14.9 (68)	1044	15.2 (539)	332.7	18.8 (1353)	154.3	18.8 (877)
TG	40, 80	989.5	12.2 (128)	5268	13.2 (572)	937.2	15.4 (1736)	443.4	15.4 (1255)
TG	50, 100	5453	14.3 (183)	Nopt (6.9%)	16.7 (1682)	4829	18.3 (3357)	1537.3	18.3 (2172)
TG	60, 120	Nopt (3.7%)	13.8 (175)	Nopt (8.6%)	14.2 (1843)	Nopt (3.2%)	16.5 (3776)	5234	16.5 (4275)
			(55)		(474)		(973)		(666)
RG	25, 150	442.3	15.0 (72)	945.2	16.2 (553)	489.1	17.1 (1296)	226.4	17.1 (924)
RG	30, 200	Nopt (0.2%)	15.5 (127)	Nopt (4.1%)	16.1 (1183)	7111	17.9 (6421)	3123	17.9 (4318)
RG	40, 120	3612	13.2 (196)	Nopt (3.3%)	14.8 (974)	3828	16.4 (3689)	1805	16.4 (3150)
RG	50, 120	Nopt (2.3%)	14.7 (188)	Nopt (8.6%)	15.6 (2158)	Nopt (2.6%)	17.2 (4523)	6006	17.2 (5298)
RG	60, 100	Nopt (3.8%)	16.2 (209)	Nopt (9.9%)	17.2 (1542)	Nopt (3.6%)	19.4 (4175)	6419	19.4 (4941)

In Table 4 we report the results obtained with (Bi-SOCP), (CL), (SS) and (ISS). In our experiments, each instance belongs to one of four graph types: series-parallel graph (SP), planar grid graph (PG), toroidal grid graph (TG) and random graph (RG). Series-parallel graphs are highly sparse while random graphs are denser graphs with $m \approx (4 \text{ to } 8) \times n$. As for each value of n, m, we have five instances, the first column of each technique in this table report the average CPU time to obtain the solution, the second column of each technique report the average gaps at root node. For the (ISS), the CPU time to calculate the bounds is included in the total CPU time to obtain the solution. For a specific technique and for the instances that this technique guarantee a solution, we note the CPU time as "Nopt" and we indicate the residual gap in parenthesis.

As can be seen from Table 1, (ISS) has the highest performance while (CL) is the least efficient technique in term of computation times. We can arrange the order of effectiveness of these techniques as (ISS)$\succ$(SS)$\succ$(Bi-SOCP)$\succ$(CL).

The main observations which arise from the results are the following :

- As discussed above, the (CL) technique does not perform well for large instances due to a large number of added variables and added constraints. We report the solution times for (CL) are worse by a factor of 2 to 4 than those of (Bi-SOCP).
- For the small instances, (SS) does not outperform (Bi-SOCP). However for the larger instances, (SS) is slightly faster than (Bi-SOCP) in spite of the fact that the number of nodes in the search tree for (Bi-SOCP) is quite reduced. A possible explanation of this is that : **a)** the computational effort required for solving each node is significant, and **b)** SOCP solvers do not enjoy the same warm-starting capabilities as simplex-based LP solvers.
- (ISS) clearly dominates the other techniques as we can see in the table, it is faster by a factor of 1.5 to 3 than (SS) in term of solution times.
- There are also instances for which CPLEX is not even capable to solve (Bi-SOCP) and (CL), but succeeds to find optimal integer solution with (SS) and (ISS). With (ISS) we can solve larger instances, e.g, $(n = 80)$ for series-parallel graphs, $(n = 70)$ for planar grid graphs, $(n = 60)$ for toroidal grid graphs and $(n = 60)$ for random graphs.
- We also report the average gaps at the root node for each solution technique. Since (CL), (SS) and (ISS) may be considered as relaxations of (Bi-SOCP), the latter provides the best gaps. Another observation is that although improved bounds for λ^{max} have been used in (ISS), the gaps at root node have not changed. However the number of nodes in the tree search is reduced in most cases, leading to the reduction of CPU times.

Finally, Table 2 shows the impact of computing the improved bounds for λ_{ui}^{max} on total CPU times for (ISS). The computation times of the bounds reported in the third column are very small as compared to the total solution times of (ISS), while the average improvement on the bounds is quite significant.

Table 2 Computation times of the bounds λ_{ui}^{max} by (8)

n,m	CPU	Percentage(%) of total CPU	average improvement on bounds(%)
25, 55	0	0	40
30, 70	0.1	0.7	39
35, 80	0.2	0.6	43
40, 75	0.4	0.7	36
45, 95	0.6	0.6	48
50, 80	0.8	0.5	45
60, 90	1.0	0.2	50
80, 130	1.8	0	43

5 Conclusion

In this paper, a stochastic version of the node weighted graph partitioning problem has been investigated. Practical application in the context of partitioning process network problem has been discussed. It has been shown that transforming an initial SOCP based formulation into quadratic constraints and applying some linearization techniques can be more efficient than solving the usual binary SOCP formulation. As a possible direction for future investigations, it would be interesting to check whether the same type of conclusion can be drawn for other combinatorial optimization problems e.g featuring knapsack constraints with random coefficients.

6 Compliance with Ethic Standards

This work was partially funded by the Gaspard Monge Program for Optimization and operations research (PGMO) supported by EDF and the Jacques Hadamard Mathematical Foundation (FMJH). The authors declare that they have no conflict of interest.

References

1. F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36(2):157–173, 1986.
2. A. Barbu and S. Zhu. Graph partition by swendsen-wang cuts. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 320–327 vol.1, Oct 2003.
3. A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, November 1998.
4. D. Bertsimas and M. Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, January 2004.
5. P. Bonami, V.H Nguyen, M. Klein, and M. Minoux. On the solution of a graph partitioning problem under capacity constraints. In A.R. Mahjoub, V. Markakis, I. Milis, and V.Th. Paschos, editors, *ISCO*, volume 7422 of *Lecture Notes in Computer Science*, pages 285–296. Springer, 2012.
6. A. Charnes and W.W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, October 1959.
7. S. Chopra and M.R. Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, 1993.
8. S. Chopra and M.R. Rao. Facets of the k-partition polytope. *Discrete Applied Mathematics*, 61(1):27 – 48, 1995.
9. W. J. Cocke. Central limit theorems for sums of dependent vector variables. *The Annals of Mathematical Statistics*, 43(3):pp. 968–976, 1972.
10. M. Deza and M. Laurent. Facets for the cut cone i. *Mathematical Programming*, 56(1-3):121–160, 1992.
11. M. Deza and M. Laurent. Facets for the cut cone ii: Clique-web inequalities. *Mathematical Programming*, 56(1-3):161–188, 1992.
12. N. Fan, Q.P. Zheng, and P.M. Pardalos. On the two-stage stochastic graph partitioning problem. In Weifan Wang, Xuding Zhu, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications*, volume 6831 of *Lecture Notes in Computer Science*, pages 500–509. Springer Berlin Heidelberg, 2011.
13. W. Feller. ber den zentralen grenzwertsatz der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 40(1):521–559, 1936.
14. C.E. Ferreira, A. Martin, C.C. de Souza, R. Weismantel, and L.A. Wolsey. The node capacitated graph partitioning problem: A computational study. *Mathematical Programming*, 81(2):229–256, 1998.
15. R. Fortet. L'algèbre de boole et ses applications en recherche operationnelle. *Trabajos de Estadistica*, 11(2):111–118, 1960.
16. A. Frangioni, A. Lodi, and G. Rinaldi. New approaches for optimizing over the semimetric polytope. *Mathematical Programming*, 104(2-3):375–388, 2005.
17. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
18. O. Goldschmidt, A. Laugier, and E.A. Olinick. Sonet/sdh ring assignment with capacity constraints. *Discrete Applied Mathematics*, 129(1):99 – 128, 2003. Algorithmic Aspects of Communication.
19. G. Karypis and V. Kumar. MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0, 2009.
20. M. Labbé and F.A. Özsoy. Size-constrained graph partitioning polytopes. *Discrete Mathematics*, 310(24):3473 – 3493, 2010.
21. J.W. Lindeberg. Eine neue herleitung des exponentialgesetzes in der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 15(1):211–225, 1922.

22. A.M. Liapunov. *Collected Works of Academician A.M. Lyapunov*. Number v. 1-2 in Collected Works of Academician A.M. Lyapunov. Translation Division, Foreign Technology Division, 1967.
23. M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(13):193 – 228, 1998. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.
24. D.P. Nguyen, M. Minoux, V.H. Nguyen, T.H. Nguyen, and R. Sirdey. Improved compact formulations for graph partitioning in sparse graphs. *Submitted to Discrete Optimization*, 2014.
25. A. Shapiro, D. Dentcheva, and A. Ruszczyski. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 2009.
26. H.D. Sherali and J.C. Smith. An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters*, 1(1):33–47, 2007.
27. M.M. Sørensen. Facet-defining inequalities for the simple graph partitioning polytope. *Discrete Optimization*, 4(2):221 – 231, 2007.
28. O. Stan, R. Sirdey, J. Carlier, and D. Nace. The robust binomial approach to chance-constrained optimization problems with application to stochastic partitioning of large process networks. *Journal of Heuristics*, 20(3):261–290, 2014.
29. W.W. Stein and T.Z. William, editors. *Applications of Stochastic Programming*. Society for Industrial and Applied Mathematics, 2005.