EURO Journal on Computational Optimization manuscript No.
(will be inserted by the editor)

Multi-weight graph partitioning problem.

D. Recalde · R. Torres · P. Vaca

Received: date / Accepted: date

Abstract In this work a multi-weight graph partitioning problem is introduced. This problem consists in partitioning an undirected graph in a fixed number of subgraphs such that multiple node weight constraints over each partition are satisfied and the total distance between nodes in the same subgraph induced by the partition is minimized. This problem generalizes several graph partitioning problems like k-way equipartition, balanced k-way partition with weight constraints, size-constrained partition, equipartition, and bisection. It arises as a subproblem of an integrated vehicle and pollster problem from a real-world application. Two Integer Programming formulations are provided and several families of valid inequalities associated to the respective polyhedra are proved. An exact algorithm based on branch and bound and cutting planes is proposed and it is tested on real-world instances.

Keywords graph partitioning · integer programming · branch & cut.

1 Introduction

The aim of partitioning arises when a set of objects characterized by attributes must be grouped into several subsets, such that some requirements must be

D. Recalde
Escuela Politécnica Nacional, Department of Mathematics, Ladrón de Guevara E11-253, Quito, Ecuador.
E-mail: diego.recalde@epn.edu.ec

R. Torres
Escuela Politécnica Nacional, Department of Mathematics, Ladrón de Guevara E11-253, Quito, Ecuador.
E-mail: ramiro.torres@epn.edu.ec

P. Vaca
Escuela Politécnica Nacional, Department of Mathematics, Ladrón de Guevara E11-253, Quito, Ecuador.
E-mail: polo.vaca@epn.edu.ec

satisfied in each group (cardinality, lower and upper bounds over quantified attributes, etc). One can think of a network where the objects correspond to nodes of the graph; attributes determine a set of weights for each node, and similarity or dissimilarity between nodes can be expressed with the inclusion of a distance function over the edges joining a pair of them. A partition of the set of nodes is a collection of non-empty subsets, pairwise disjoint, such that their union is the original set. If each subset of a partition satisfies constraints involving the attributes of the nodes and some objective function is optimized, then a multi-weight graph partitioning problem arises.

This problem appears as a sub-problem in the following real-world application: The National Statistics Bureau of Ecuador (INEC) carries out monthly polls to monitor the behavior of consumer prices for basic commodities, which are collected from a fixed set of stores known a-priori and located around a city in Ecuador [10]. Each store has to be visited once a month by a pollster, who registers prices of some commodities previously specified by INEC. After the information is collected at the store, each pollster moves on to the next scheduled store. The polls must be performed on a fixed number of days and the available number of pollsters of the Bureau office varies per day. Thus, the task associated to the sub-problem in the previous application consists of partitioning the set of stores in a fixed number of subsets where each subset represents the stores to be visited in a day. The requirements on each subset are diverse and look forward to preserve some homogeneity for the subsets according to requirements as: number of stores, working time, pollster waiting time, among others. The aim of the sub-problem, concerning the stores partition, consists of minimizing the total distance among stores on each subset of the partition.

The study of graph partitioning problems started in the early seventies with the seminal work of Kernighan and Lin [13]. Here, the authors partitioned a graph into subsets of given sizes and they proposed a heuristic method to solve it. Christofides and Brooker[4] studied the bipartition problem where a tree search method imposing an upper bound on the size of nodes in one subset is considered. Carlson and Nemhauser [2] formulated a quadratic program for partitioning a graph in at most k subsets with no bound on the size of each subset. Gröstchel and Wakabayashi [8] introduced the clique partitioning problem on a complete graph, where the problem is studied from a polyhedral point of view. Rao and Chopra [3] presented the k-partitioning problem where k, the number of subsets in the partition, is given a-priori; the authors described integer programming formulations on a connected graph together with several facets and valid inequalities for the associated polytopes. Ferreira et al. [7] introduced capacity constraints on the sum of node weights in each subset of the partition. In similar way, Labbé and Özsoy [14] formulated the problem as the clique partition problem including upper and lower bounds on the size of the cliques and a competitive branch-and-cut algorithm was implemented. In regard to applications, the graph partitioning problem is widely applied in distribution of work to processors in parallel computing [11], VLSI circuit design [12], mobile wireless communications [5], or sports team realignment

[16]. In Buluc et al.[1], a complete survey of applications and recent advances in graph partitioning can be found.

The paper is organized as follows. In Section 2 some formulations are described, where reduced models preserving the strength of the relaxation are identified. Section 3 includes some techniques attending to strengthen the initial formulations. In Section 4 multiple valid inequalities for the multi-weight graph partitioning problem are introduced and computational experiments are discussed in Section 5.

2 Notation and Integer programming formulations

Let $G = (V, E)$ be an undirected graph with $V = \{1, \dots, n\}$ the set of nodes, $E = \{\{i, j\} : i, j \in V, i \neq j\}$ the set of edges, $d : E \to \mathbb{R}^+$ a distance function, and $k \geq 2$ an integer number. Let $\{V_1, V_2, \dots, V_k\}$ be a k-partition of the set V, i.e., $V_i \cap V_j = \emptyset$ for all $i \neq j$, $\bigcup_{c=1}^{k} V_c = V$, and $V_c \neq \emptyset$ for all $c \in [k]$, where $[k]$ denotes the set $\{1, \dots, k\}$. Moreover, a τ-dimensional vector $(w_i^1, \dots, w_i^\tau) \geq 0$ of weights associated to each node i is defined. Each $t \in [\tau]$ is called an *attribute*. Similarly, lower and upper bound vectors $W_L^c, W_U^c \in \mathbb{R}_+^\tau$, with $W_L^c \leq W_U^c$, are introduced for all $c \in [k]$. For the sake of simplicity, $w^t(S) = \sum_{i \in S} w_i^t$, for $S \subset V$ is defined.

The multi-weight graph partitioning problem ($\mathcal{MWGP}$) consists of finding a k-partition $\{V_1, V_2, \dots, V_k\}$ such that the *requirements*

$$W_L^{ct} \leq w^t(V_c) \leq W_U^{ct}, \qquad \forall c \in [k],\ t \in [\tau] \tag{1}$$

are satisfied and the total cost of the edges with end nodes in the same subset of the partition is minimized.

Notably, to the extent of our knowledge, the multi-weight graph partitioning problem has not been reported in the literature. In fact, other partitioning problems can be obtained from $\mathcal{MWGP}$ by fixing parameters adequately. For instance, if multi-weight constraints (1) are suppressed, the problem turns out to be a k-partitioning problem [3]. Similarly, if $\tau = 1$, $n \equiv 0 \pmod{k}$, the weight of each node is fixed to one, and the lower and upper bound are equal to n/k, the problem becomes the so-called *k-way equipartition problem* [15]. If lower and upper bounds over the size of the subsets in the partition are imposed, then the size-constrained graph partitioning problem appears [14]. Moreover, when cardinalities $n_1, \dots, n_k$ for each subset of the partition are given a-priori, and by fixing $W_L^1 = W_U^1 = n_1, \dots, W_L^k = W_U^k = n_k$, the general graph partitioning problem arises [6]. In the same context, if $k = 2$ and $n_1 = n_2$, then the equipartition problem is obtained and if $n_1 \neq n_2$ the bisection problem on a graph comes up [4,13]. On the other hand, if $\tau = 2$, where the first requirement corresponds to the cardinality and the second one corresponds to a positive weight on each subset of the partition, the problem becomes the balanced $k-$way partitioning problem with weight constraints [17].

It is well known that all these problems are $\mathcal{NP}$-hard and thus there is a remote possibility of finding a polynomial time algorithm to solve the $\mathcal{MWGP}$ to optimality. It is also known that approaches based on Integer Linear Programming have proven to be one of the best tools to solve these kind of hard problems exactly.

2.1 First IP Formulation

This formulation considers two sets of binary variables: the first one related to nodes and the second one associated to edges. Thus, let y_i^c be the variable that takes the value 1 if the node $i \in V$ belongs to subset V_c, for all $c \in [k]$, and 0 otherwise. Moreover, $x_{ij} = 1$ if the edge $\{i, j\} \in E$ corresponds to a pair of nodes in the same subset of a partition and $x_{ij} = 0$ otherwise. Then, the $\mathcal{MWGP}$ can be formulated as $(\mathcal{F}_1)$:

$$\min \sum_{\{i,j\} \in E} d_{ij} x_{ij} \tag{2}$$

$$\sum_{c \in [k]} y_i^c = 1, \qquad \forall i \in V \tag{3}$$

$$y_i^c + y_j^c - x_{ij} \leq 1, \qquad \forall \{i, j\} \in E,\ c \in [k], \tag{4}$$

$$W_L^{ct} \leq \sum_{i \in V} w_i^t y_i^c \leq W_U^{ct}, \qquad \forall\ c \in [k], t \in [\tau] \tag{5}$$

$$y_i^c \in \{0, 1\}, \qquad \forall i \in V, c \in [k], \tag{6}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall \{i, j\} \in E, \tag{7}$$

The objective function (2) is the total edge cost of the subgraphs $(V_c, E(V_c))$, for all $c \in [k]$. Constraints (3) indicate that each node must belong exactly to one subset, constraints (4) establish that if two nodes $i, j \in V$ are assigned to the subset V_c, then the edge $\{i, j\} \in E$ belongs to $E(V_c)$. Finally, constraints (5) guarantee that the τ-dimensional requirements over each subset in the partition are satisfied. The formulation can be considered as a generalization of [3].

2.2 Second IP Formulation

The polyhedron associated to $\mathcal{F}_1$ can be projected into an edge-variable space through a k-augmenting graph which is defined as follows.

Definition 1 Let $G = (V, E)$ be an undirected graph with n nodes, a distance function $d : E \to \mathbb{R}_+$ over the edges, a $\tau-$dimensional weight vector $(w_i^1, \ldots, w_i^\tau)$ on each node $i \in V$ and a fixed integer number $k \geq 2$. The k-augmenting graph is the pair $G^k = (V^k, E^k)$, where $V^k = V \cup \mathbb{A}$, $\mathbb{A} = \{n + 1, n + 2, \ldots, n + k\}$ is a set of artificial nodes (one for each subset of the partition), $E^k = E \cup E_1 \cup E_2$, $E_1 = \{\{i, j\} : i \in V, j \in \mathbb{A}\}$

and $E_2 = \{\{i, j\} : i, j \in \mathbb{A}, i < j\}$. Edges $e \in E$ maintain the original distance d_e, edges in E_2 have sufficiently large distance $M > 0$ and edges in E_1 have distance equal to 0. Finally, original nodes mantain the vector of weights $(w_i^1, \dots, w_i^\tau)$ and $w_j^t = 0$ for all $j \in \mathbb{A}$ and $t \in [\tau]$.

The k-augmenting graph $G^k = (V^k, E^k)$ allows us to formulate $\mathcal{MWGP}$ using the classical triangular inequalities ($\mathcal{F}_2$):

$$\min \sum_{\{i,j\} \in E^k} d_{ij} x_{ij} \tag{8}$$

$$+ x_{ij} + x_{jl} - x_{il} \leq 1, \qquad \forall 1 \leq i < j < l \leq |V^k|, \tag{9}$$

$$+ x_{ij} - x_{jl} + x_{il} \leq 1, \qquad \forall 1 \leq i < j < l \leq |V^k|, \tag{10}$$

$$- x_{ij} + x_{jl} + x_{il} \leq 1, \qquad \forall 1 \leq i < j < l \leq |V^k|, \tag{11}$$

$$\sum_{c \in \mathbb{A}} x_{ic} = 1, \qquad \forall i \in V, \tag{12}$$

$$W_L^{ct} \leq \sum_{i \in V} w_i^t x_{i,n+c} \leq W_U^{ct}, \qquad \forall\, c \in [k],\ t \in [\tau], \tag{13}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall \{i, j\} \in E^k. \tag{14}$$

As it is well known, this formulation requires a complete graph[8]. Thus, if G^k is not complete, artificial edges with distance equal to zero are added adequately to this graph.

The binary variables x_{ij} are defined in the same way as the previous formulation for every $\{i, j\} \in E^k$ as well as the objective function. Constraints (9)-(11) are the classical triangular inequalities; and constraints (12) ensure that every node i belongs to exactly one subset V_c of the partition. Finally, inequalities (13) impose lower and upper bounds for the total weight of each subset for every requirement.

The convex hull of the set of points $(x(E), y) \in \{0, 1\}^{|E| \times k|V|}$ satisfying (3)-(7) defines the polytope denoted by P^{xy}. In similar way, the polytope $P^x = \text{conv}(\{x(E^k) \in \{0, 1\}^{|E^k|} : (9) - (14) \text{ are fullfilled}\})$ is introduced. The following Theorem establishes a correspondence between the set of points of both polytopes.

Theorem 1 *$x(E^k) \in P^x$ with objective value strictly less than M if and only if $(x(E), y) \in P^{xy}$.*

Proof Let $x(E^k)$ be a point in P^x with objective value strictly less than M. Then, $x(E_2) = 0$ and $(x(E), x(E_1)) \in P^{xy}$ where $y_i^c = x_{i,n+c}$ for all $c \in [k]$ and $i \in V$. The other implication follows immediately by fixing variables $x(E_2) = 0$ and $x(E_1) = y$.

Note that the multi-weight constraints could make empty the polytopes of both formulations. In this sense, a necessary condition for feasibility on multi-weights is established in the following result:

Theorem 2 *A necessary condition for the feasibility of the multi-weight graph partitioning problem is:*

$$\max_{t\in[\tau]}\left\{\left\lceil\frac{\sum_{i\in V} w_i^t}{max_{c\in[k]}\{W_U^{ct}\}}\right\rceil\right\} \leq k \leq \min_{t\in[\tau]}\left\{\left\lfloor\frac{\sum_{i\in V} w_i^t}{\min_{c\in[k]}\{W_L^{ct}\}}\right\rfloor\right\}$$

Proof From constraints (5) of formulation $\mathcal{F}_1$, the following constraints are given for each $t \in [\tau]$:

$$\sum_{c\in[k]} W_L^{ct} \leq \sum_{c\in[k]}\sum_{i\in V} w_i^t y_i^c \leq \sum_{c\in[k]} W_U^{ct},$$

Since $V_c = \{i \in V : y_i^c = 1\}$, the last expression can be written, for each $t \in [\tau]$, as:

$$k \min_{c\in[k]}\{W_L^{ct}\} \leq \sum_{c\in[k]}\sum_{i\in V_c} w_i^t \leq k \max_{c\in[k]}\{W_U^{ct}\},$$

$$k \min_{c\in[k]}\{W_L^{ct}\} \leq \sum_{i\in V} w_i^t \leq k \max_{c\in[k]}\{W_U^{ct}\},$$

Since the previous condition must be satisfied for each $t \in [\tau]$, then the result follows.

3 Preprocessing

The preprocessing techniques described in this section attend to strengthen the previous formulations of the graph partitioning problem. In this context, variable reduction techniques and methods for reducing the number of constraints are reported.

3.1 Constraint reduction

For formulation ($\mathcal{F}_2$), one can identify some redundant triangular inequalities. Thus, consider that three nodes $i, j, p \in V$ are assigned to subset l. It implies that the associated edges $\{i, l\}$, $\{j, l\}$ and $\{p, l\}$ must be included in the solution and the corresponding variables are equal to one. For triangular inequalities, if edges $\{i, l\}$ and $\{j, l\}$ belong to the same subset of a partition, then $\{i, j\}$ is obviously in the same partition. In similar way, considering edges $\{j, l\}, \{p, l\}$ and $\{i, l\}, \{p, l\}$ determine that edges $\{j, p\}$ and $\{i, p\}$ are included in the solution. Hence, the triangular inequalities associated to edges $\{i, j\}, \{j, p\}$ and $\{i, p\}$ are not needed anymore in our formulation. Therefore, observe that these constraints reduction allows to apply this formulation to non-complete graphs as none of the triangular inequalities for three nodes in V appears.

The reduced model can be reformulated as follows ($\mathcal{FR}_2$):

$$\min \sum_{\{i,j\}\in E} d_{ij}x_{ij} \tag{15}$$

$$+ x_{ij} + x_{jl} - x_{il} \le 1, \qquad \forall \{i,j\} \in E,\ l \in \mathbb{A} \tag{16}$$

$$+ x_{ij} - x_{jl} + x_{il} \le 1, \qquad \forall \{i,j\} \in E,\ l \in \mathbb{A} \tag{17}$$

$$- x_{ij} + x_{jl} + x_{il} \le 1, \qquad \forall \{i,j\} \in E,\ l \in \mathbb{A} \tag{18}$$

$$\sum_{l\in\mathbb{A}} x_{jl} = 1, \qquad \forall j \in V \tag{19}$$

$$W_L^{ct} \le \sum_{j\in V} w_j^t x_{jl} \le W_U^{ct}, \qquad \forall\, l \in \mathbb{A},\ t \in [\tau], \tag{20}$$

$$x_{ij} \in \{0,1\}, \qquad \forall \{i,j\} \in E^k, \tag{21}$$

It is important to remark that, from now and on, only formulation $\mathcal{F}_1$ and $\mathcal{FR}_2$ are used for theoretical and computational purposes.

3.2 Variables reduction

Initially, a pre-fixing process tries to reduce the number of variables considered in the initial formulation. The idea follows from the simple fact that if we focus on one node, say node 1, it must belong to some subset of the partition, say subset 1, and therefore such node can not be included in the remaining sets $c \in \{2, 3, \dots, k\}$. Taking a second node, say node 2, it could belong to subset 1 or another subset (subset 2 for example), and then such a node cannot belong to the remaining subsets $c \in \{3, 4, \dots, k\}$. Up to now, the variables fixed to zero can be excluded from the branching process without losing any feasible solution. Extending this idea, the $k-$th node will belong to one of the first k subsets and the number of node variables that will not be part of the branching has order $\mathcal{O}(k^2)$.

Hence, for the formulation $\mathcal{F}_1$ the pre-fixing process is determined as follows:

- $y_i^c = 1$, for all $i = c = 1$,
- $y_i^c \in \{0, 1\}$, for all $i = 2, \dots, k$, $c \in [i]$,
- $y_i^c = 0$, for all $i \in [k]$, $c = i + 1, \dots, k$.

For the formulation $\mathcal{FR}_2$ the pre-fixing process is stated:

- $x_{i,n+c} = 1$, for all $i = c = 1$,
- $x_{i,n+c} \in \{0, 1\}$, for all $i = 2, \dots, k$, $c \in [i]$,
- $x_{i,n+c} = 0$, for all $i \in [k]$, $c = i + 1, \dots, k$.

Moreover, a branching rule is presented. It reduces the size of the search tree and increases the speed of the solution process. The branching procedure uses only variables associated to nodes, since if two specific nodes $i, j \in V$ are assigned to subset c, for any $c \in [k]$, then automatically the edge $\{i, j\}$ must be included in the subgraph. Thus, in formulation $\mathcal{F}_1$ given $y_i^c = y_j^c = 1$, then $x_{ij} = 1$ is propagated. Moreover, if $y_i^c = 1$ and $y_j^c = 0$ or $y_j^d = 1$ for any $c \in [k]$ and $d \neq c$, then it implies that $x_{ij} = 0$. In similar way for formulation $\mathcal{F}_2$ and $\mathcal{FR}_2$, if $x_{i,l} = x_{j,l} = 1$ for any $l \in \mathbb{A}$, then for triangular inequalities $x_{ij} = 1$ is propagated. The second implication indicates that if $x_{i,l} = 1$ and $x_{j,l} = 0$ or $x_{j,p} = 1$ for any $p \in \mathbb{A}$ and $p \neq l$, then $x_{ij} = 0$ is hold.

4 Valid inequalities

In this section, a variety of classes of valid inequalities is stated. The polytopes defined by the convex hull of integer solutions of the formulations is uniquely determined by the parameters k, lower and upper bounds W_L^{ct}, W_U^{ct} for $c \in [k]$, $t \in [\tau]$, and w_i^t for $i \in V$, $t \in [\tau]$.

Theorem 3 *Let $G = (V, E)$ the graph associated to the second formulation and let $G^k = (V^k, E^k)$ be its augmenting graph. For every two disjoint subsets $S \subseteq V^k$, $R \subseteq V^k$, the inequality induced by S and R:*

$$\sum_{\substack{\{i,j\} \in E^k: \\ i \in S, j \in R}} x_{ij} - \sum_{\{i,j\} \in E^k(S)} x_{ij} - \sum_{\{i,j\} \in E^k(R)} x_{ij} \leq \min\{|S|, |R|\}$$

is valid for $\mathcal{P}^x$

Proof First, we add dummy edges to G^k with weights equal to zero in order to obtain a complete graph. As any feasible solution of the $\mathcal{MWGP}$ is a feasible solution of the clique partitioning problem, then the result follows from the well-known 2-partition inequality of [8].

The following corollary follows from the fact that the value of the variables associated to every two artificial nodes in the set $\mathbb{A}$ are equal to zero.

Corollary 1 *Let $G = (V, E)$ the graph associated to the second formulation and let $G^k = (V^k, E^k)$ be its augmenting graph. For every two disjoint subsets $S \subseteq V$, $R \subseteq \mathbb{A}$, the inequality induced by S and R:*

$$\sum_{\substack{\{i,j\} \in E^k: \\ i \in S, j \in R}} x_{ij} - \sum_{\{i,j\} \in E^k(S)} x_{ij} \leq \min\{|S|, |R|\}$$

is valid for $\mathcal{P}^x$

Considering the relation between variables x_{ij}, $i \in V, j \in \mathbb{A}$ and variables y_i^c, $i \in V, c \in [k]$, the following corollary is stated.

Corollary 2 *Let $G = (V, E)$ the graph associated to the first formulation. For every pair of subsets $S \subseteq V$, $R \subseteq [k]$, the inequality induced by S and R:*

$$\sum_{i \in S, c \in R} y_i^c - \sum_{\{i,j\} \in E(S)} x_{ij} \leq \min\{|S|, |R|\}$$

is valid for $\mathcal{P}^{xy}$

Definition 2 A subset $S \subseteq V$ is called a cover of a subset V_c, for some $c \in [k]$, if $w^t(S) > W_U^{ct}$ for all attribute $t \in [\tau]$.

Definition 3 The subset $S \subseteq V$ is called a minimal cover of a subset V_c, for some $c \in [k]$, if S is a cover and for at least one requirement t, $w^t(S \setminus \{j\}) \leq W_U^{ct}$, with $j \in S$.

Definition 4 The subset $S \subset V$ is called a global cover of graph G if S is a cover for all V_c, $c \in [k]$. A global cover is minimal if S is minimal for all V_c, $c \in [k]$.

The following results provide families of valid inequalities for P^{xy} and P^x.

Theorem 4 *Let $S = \{i, j\} \subset V$, such that the edge joining nodes i and j belongs to E. If the set S is a global cover, then $x_{ij} = 0$ is a valid equation for P^{xy} and P^x.*

Proof Observe that if S is global cover, then i and j cannot belong to the same subset V_c, for any $c \in [k]$

Theorem 5 *For any cycle $\mathcal{C}$ in G such that $V(\mathcal{C})$ is a global cover, then the inequality:*

$$\sum_{\{i,j\} \in E(\mathcal{C})} x_{ij} \leq |E(\mathcal{C})| - 2$$

is valid for P^{xy} and P^x

Proof $V(\mathcal{C})$ becomes a feasible subset of any partition if and only if at least one node (two edges) is dropped from $V(\mathcal{C})$.

A generalization of the above result is the so called q-cover. A node set $S \subset V$ is a q-cover, $q \geq 1$ and integer, if $w^t(S) > q \max_{c \in [k]} \{W_U^{ct}\}$, for all requirement $t \in [\tau]$.

Theorem 6 *For any cycle $\mathcal{C}$ in G such that $V(\mathcal{C}) \in V$ is a q-cover, then the inequality:*

$$\sum_{\{i,j\} \in E[\mathcal{C}]} x_{ij} \leq |E(\mathcal{C})| - q - 1$$

is valid for P^{xy} and P^x.

Proof $V(\mathcal{C})$ becomes a feasible subset of any partition if and only if at least q consecutive nodes in the cycle ($q + 1$ edges) are dropped from $V(\mathcal{C})$.

A similar result follows when the subgraph $(S, E(S))$ is a tree:

Corollary 3 *Let $S \subseteq V$ be a minimal global cover and the induced subgraph $(S, E(S))$ is a tree. Then,*

$$\sum_{\{i,j\} \in E(S)} x_{ij} \leq |E(S)| - 1$$

is a valid inequality for P^{xy} and P^{x}.

More generally, for every $S \subset V$, and any induced subgraph $(S, E(S))$, the following inequalities are valid.

Theorem 7 *Let $S \subset V$ be a minimal cover of a subset V_c for some $c \in [k]$. Then, the following cover inequalities:*

$$\textstyle\sum_{i \in S} |\delta(i)| y_i^c \leq 2|E(S)| - \min_{i \in S}\{|\delta(i)|\}$$

are valid for $\mathcal{P}^{xy}$, and

$$\textstyle\sum_{i \in S} |\delta(i)| x_{i,n+c} \leq 2|E(S)| - \min_{i \in S}\{|\delta(i)|\}$$

are valid for $\mathcal{P}^{x}$.

Proof Observe that the set $S \setminus \{v\}$ for some $v \in S$ belongs to any feasible solution. In the worst case this is true by choosing the node with minimum degree.

The following theorem extends the previous result for minimal global covers.

Theorem 8 *Let $S \subseteq V$ be a minimal global cover and $(S, E(S))$ the subgraph induced by S. The inequality:*

$$\sum_{\{i,j\} \in E(S)} x_{ij} \leq |E(S)| - \min_{i \in S}\{|\delta(i)|\}$$

is valid for $\mathcal{P}^{xy}$ and $\mathcal{P}^{x}$.

Theorem 9 *Let $c \in [k]$ be a fixed number and $S \subset V$ with $w^t(S) \leq W_U^{ct}$ for all $t \in [\tau]$. Moreover, let $r_t = W_U^{ct} - w^t(S)$ and $\mathcal{X} = \{i \in V \setminus S : w_i^t > r_t, t \in [\tau]\}$ be a non-empty set. The (S, c)-family of inequalities with respect to S*

$$\sum_{i \in S} w_i^t y_i^c + \sum_{j \in \mathcal{X}} (w_j^t - r_t) y_j^c \leq w^t(S), \qquad \forall t \in [\tau],$$

is valid for P^{xy} and,

$$\sum_{i \in S} w_i^t x_{i,n+c} + \sum_{j \in \mathcal{X}} (w_j^t - r_t) x_{j,n+c} \leq w^t(S), \qquad \forall t \in [\tau],$$

is valid for $\mathcal{P}^{x}$.

Proof To prove this result, two cases are considered:

– If $\sum_{j\in\mathcal{X}} y_j^c = 0$, then the family of inequality are trivially valid.
– If $\sum_{j\in\mathcal{X}} y_j^c > 0$, then for each $t \in [\tau]$

$$\begin{aligned}\sum_{i\in S} w_i^t y_i^c + \sum_{j\in\mathcal{X}} (w_j^t - r_t) y_j^c &= \sum_{i\in S} w_i^t y_i^c + \sum_{j\in\mathcal{X}} w_j^t y_j^c - \sum_{j\in\mathcal{X}} r_t y_j^c \\ &\leq \sum_{i\in V} w_i^t y_i^c - \sum_{j\in\mathcal{X}} r_t y_j^c \\ &\leq W_U^{ct} - r_t \\ &= w^t(S)\end{aligned}$$

The validity of the second family of inequalities for $\mathcal{P}^x$ is proven in a similar way due to the correspondence between the variables y_i^c and $x_{i,n+c}$.

The previous family of inequalities can be rewritten in terms of the edge variables x_{ij} which appear in both formulations.

Theorem 10 *Let $S \subset V$ with $w^t(S) \leq \min_c\{W_U^{ct}\}$ for each $t \in [\tau]$ and $\mathcal{X} = \{j \in V \setminus S : w_j^t > r_t, \forall t \in [\tau]\} \neq \emptyset$, where $r_t = min_c\{W_U^{ct}\} - w^t(S)$. For each fixed node $i \in S$, the (S, i)-family of inequalities*

$$w_i^t + \sum_{l\in S\setminus\{i\}} w_l^t x_{il} + \sum_{j\in\mathcal{X}} (w_j^t - r_t) x_{ij} \leq w^t(S), \quad \forall t \in [\tau]$$

is valid for P^{xy} and $\mathcal{P}^x$.

5 Computational experiments

Some computational experiments with our IP formulations and valid inequalities are carried out in this section. A set of tests are fulfilled and they consist of solving the formulations $\mathcal{F}_1$ and $\mathcal{FR}_2$ and combining them with valid inequalities in different ways. For the first formulation the valid inequalities 2-part$_{xy}$ (Corollary 2), cycle (Theorem 5), subgraph (Theorem 8), and upper-bound (Theorem 10) are included, meanwhile the second reduced formulation considers 2-part$_x$ (Corollary 1), cycle (Theorem 5), subgraph (Theorem 8), and upper-bound (Theorem 10) as valid inequalities.

The corresponding instances are samples of real-world data arising from an application of the INEC introduced in Section 1. The case study is focused in Guayaquil, the second most populous city of Ecuador. Given two positive integer numbers n and k, with $n > k$, a sample instance consists of n points randomly chosen from a population of 820 stores together with working times for collecting data and waiting times spent by pollsters in queues at each selected store. The distance between two stores is the pedestrian travel time. Thus, one can associate the location of stores with the nodes of a graph, the distance function with pedestrian travel times between stores, and a 2-dimensional vector of weights at each node (the first component contains working times and the second one corresponds to waiting times). Moreover, upper and lower bounds $W_L^t = \mu^t(n/k) - 2\sigma^t$ and $W_U^t = \mu^t(n/k) + 2\sigma^t$, for all $t \in \{1, 2\}$ are specified. In latter formulas, μ^t and σ^t are the average and standard deviation of weights of all nodes associated to attribute t, for $t \in \{1, 2\}$. For each pair $(n, k) \in \{(20, 3), (30, 4), (40, 5), (50, 6), (55, 6)\}$, five instances are generated according to the method described before.

The IPs were solved using the integer programming solver Gurobi 8.10 [9] and the C++ programming language interface. All the experiments were performed on an Intel Core i7 3.60 GHz with 8 GB RAM running Ubuntu 14.01. The computation time is limited to 2000 seconds for every instance.

The first experiments consist in verifying the efficiency of the variable reduction procedure explained in the preprocessing phase (Section 3.2). Both IP formulations are solved using Gurobi (cuts are disabled) with and without the aforementioned process.

Table 1 summarizes the results for $\mathcal{F}_1$ and Table 2 reports the results obtained for $\mathcal{FR}_2$. The organization of these tables is as follows: first column displays the pair (n, k) which describes the number of nodes and the number of subsets to be constructed in the instance; columns 2 to 5 report the objective function value, the optimality gap, the CUP time and the number of B&B nodes evaluated in the optimization process for the original formulation, respectively. The remaining columns show the objective function value, the optimality gap, the CPU time and the number of B&B nodes for the formulation including the preprocessing phase.

For the first formulation, instances of size $n = 20$ are solved up to the optimality in an average time of 1.94 seconds and for instances of size $n = 30$ the optimality is obtained in an average CPU time of 440.64 seconds. However, if the preprocessing phase is included, the CPU time for instances of size $n = 20$ are reduced to 0.51 seconds and for instances of size $n = 30$ the average CPU time is 30.06 seconds. In the remaining instances, the preprocessing routine reduces the optimality gap in 27, 58% in average. Moreover, one can see that in all instances the number of evaluated nodes in the B&B process is drastically reduced. These results are supported with experiments on the reduced formulation $\mathcal{FR}_2$ which are shown in Table 2. Instances of size $n = 20$ are solved up to the optimality in an average time of 3.67 seconds and 1.40 seconds when preprocessing routine is included. In this formulation, the third instance of size $n = 30$ is not solved optimally, but if the preprocessing phase is included in the optimization process, all of them are solved up to the optimality in an average CPU time of 74.82 seconds. As in the first formulation, the number of B & B nodes and the optimality gap are reduced.

The second experiment consists on working with valid inequalities presented in Section 4. The performance of the IP formulations including different combinations of separation routines is compared. Each family of valid inequalities is enumerated exhaustively, and at most 100 valid inequalities are added at each node of the branching procedure when those are violated by a factor of 0.1. In the overall Branch & Cut process at most 5000 valid inequalities are included.

	$\mathcal{F}_1$				$\mathcal{F}_1$ + preprocessing			
Instance	Obj	Gap	Time	Nodes	Obj	Gap	Time	Nodes
(20,3)	1934.69	0.00	1.99	8042	1934.69	0.00	0.63	596
	1756.26	0.00	3.34	18579	1756.26	0.00	0.71	1123
	1364.55	0.00	1.66	4338	1364.55	0.00	0.44	181
	2051.81	0.00	1.56	6283	2051.81	0.00	0.41	794
	1019.87	0.00	1.13	3021	1019.87	0.00	0.36	206
(30,4)	3324.67	0.00	158.80	332336	3324.67	0.00	18.94	19619
	2242.02	0.00	265.41	592512	2242.02	0.00	38.52	48053
	3081.71	0.00	1043.76	2256521	3081.71	0.00	80.23	105183
	2440.49	0.00	166.08	258705	2440.49	0.00	13.52	9693
	1909.96	0.00	569.17	1266582	1909.96	0.00	19.11	25457
(40,5)	3827.56	58.96	2000.00	1249639	3788.55	36.53	2000.10	901763
	3017.60	64.64	2000.00	1467655	3043.55	32.87	2000.08	843209
	3343.26	60.92	2000.00	1587761	3342.41	24.40	2000.38	1038970
	4311.32	66.35	2000.00	1315401	4374.85	27.75	2000.29	909239
	2661.67	58.76	2000.00	1104112	2661.67	9.38	2000.01	980402
(50,6)	4322.05	80.17	2000.00	558163	4260.06	53.31	2000.55	281223
	4932.87	78.82	2000.00	462737	5022.38	69.53	2000.62	316469
	3837.74	74.60	2000.00	568476	3790.46	51.01	2000.56	355945
	5312.07	85.19	2000.00	459290	5287.28	56.74	2001.23	368407
	4046.14	82.79	2000.00	550554	4168.18	61.44	2001.14	272481
(55,6)	5183.67	90.49	2000.00	382912	5474.30	75.25	2000.72	205723
	3944.51	86.93	2000.00	441375	4007.84	66.93	2001.48	194903
	5080.74	92.12	2000.00	447422	5280.94	76.42	2000.13	184141
	5036.59	91.93	2000.00	366850	5429.90	76.70	2000.82	158751
	5196.04	83.76	2000.00	276016	4918.74	62.94	2000.72	226953

Table 1 Solving $\mathcal{F}_1$ using only Gurobi.

The separation routines for formulation $\mathcal{F}_1$ are performed on each node depending on the current fractional solution denoted by (y^*, x^*). The separation of 2-part$_{xy}$ inequalities is inspired by the procedure described by [8], where the following procedure is repeated for each $c \in [k]$: first, compute $W := \{i \in V : 0 < (y_i^c)^* < 1\}$. If $|W| \geq 5$, then an arbitrary node $w \in W$ is picked and the set $S := \{w\}$ is defined. For the remaining nodes $i \in W \setminus \{w\}$, the set S is updated by $S = S \cup \{i\}$ if $x_{ij}^* = 0$ for all $j \in S$. Then, for S and $R = \{c\}$, check if the Corollary 2 is violated and, in that case, add the corresponding inequality as a cut to the current LP. Using the equivalence of variables y_i^c with $x_{i,n+c}$, the routines can be extended for $\mathcal{FR}_2$.

Regarding cycle, subgraph and upper bound inequalities, these inequalities are included as exhaustively enumerated sets of them, each one corresponding to sets of 4 and 5 nodes, as these are frequently violated and enumerating them is not a time consuming process. The following separation routines are used in both formulations. Thus, for each $\ell \in \{4, 5\}$:

- Cycle: For every subset $\mathcal{C}$ composed by ℓ nodes in V, the τ-dimensional vector $(w^1(\mathcal{C}), \ldots, w^\tau(\mathcal{C}))$ is computed. If $w^t(\mathcal{C}) > W_U^{ct}$ for all $c \in [k]$, $t \in [\tau]$, then for every cycle comformed by all nodes in $\mathcal{C}$, check if the inequality associated to Theorem (5) is violated. If that is the case, then the corresponding cycle cut is added in the LP.

	$\mathcal{FR}_2$				$\mathcal{FR}_2$ + preprocessing			
Instance	Obj	Gap	Time	Nodes	Obj	Gap	Time	Nodes
(20,3)	1934.69	0.00	3.35	5073	1934.69	0.00	1.92	3948
	1756.26	0.00	4.37	7839	1756.26	0.00	3.23	8736
	1364.55	0.00	2.58	3403	1364.55	0.00	0.64	906
	2051.81	0.00	3.28	4630	2051.81	0.00	0.69	676
	1019.87	0.00	4.78	5706	1019.87	0.00	0.52	871
(30,4)	3324.67	0.00	628.31	475931	3324.67	0.00	43.55	18773
	2242.02	0.00	777.96	508718	2242.02	0.00	86.50	39981
	3081.71	21.42	2000.00	1273517	3081.71	0.00	160.33	95873
	2440.49	0.00	770.70	424767	2440.49	0.00	36.39	13259
	1909.96	0.00	863.91	558107	1909.96	0.00	47.36	30171
(40,5)	3858.43	64.95	2000.00	348056	3956.12	47.32	2000.02	384035
	3094.83	69.45	2000.08	436075	3017.60	37.84	2000.23	367297
	3342.41	71.04	2000.00	362934	3342.41	35.09	2000.57	394486
	4355.61	72.98	2000.00	401591	4311.32	31.97	2000.07	382282
	2661.67	66.01	2000.00	345684	2661.67	20.24	2000.26	325669
(50,6)	4831.33	84.25	2000.00	143530	4216.45	56.69	2001.57	116306
	5306.78	92.26	2000.00	85963	4903.67	73.61	2000.86	92877
	4320.77	83.96	2000.01	125828	3910.31	55.83	2000.52	119317
	5630.18	90.86	2000.00	124239	5818.08	66.56	2000.59	147153
	4469.82	91.49	2000.00	174929	4645.66	71.70	2000.43	116693
(55,6)	5600.83	95.72	2000.00	66639	5511.26	80.43	2001.99	69729
	4113.25	94.03	2000.00	83412	4965.11	79.60	2001.48	69248
	4613.29	95.06	2000.00	79806	5300.93	82.86	2000.25	70413
	5828.06	95.87	2000.01	99842	4684.62	78.26	2002.11	52771
	5528.97	88.64	2000.00	68025	5112.37	70.88	2000.16	70361

Table 2 Solving $\mathcal{FR}_2$ using only Gurobi.

– Subgraph: Using the similar idea of the previous item , for every subset S in V with $|S| = \ell$, the τ-dimensional vector $(w^1(S), \dots, w^\tau(S))$ is computed. If $w^t(S) > W_U^{ct}$ for all $c \in [k]$ and $t \in [\tau]$ and the inequality associated to Theorem (8) is not hold, then the subgraph inequality is added in the current LP.
– Upper bound: For every set S composed by ℓ nodes in V, the τ-dimensional vector $(w^1(\mathcal{C}), \dots, w^\tau(\mathcal{C}))$ is computed. If $w^t(S) \leq \min_{c\in[k]}\{W_U^{ct}\}$, for all $t \in [\tau]$, then a τ-dimensional vector r is computed, where $r_t = \min_{c\in[k]}\{W_U^{ct}\} - w^t(C)$. Now, each node $j \in V \setminus S$ satisfying that $w_j^t > r_t$ is included in the subset $\mathcal{X}$. If $|\mathcal{X}| > 0$, for every node $i \in S$, if the associated inequality in Theorem (10) is violated, then such (S, i) inequality is included as a cut at the LP.

The effectiveness of the valid inequalities is measured in terms of the reduction of the number of B&B nodes and the optimality gap. Observe that for instances $(30, 4), (40, 5), (50, 6)$ and $(55, 6)$ the optimal solution was not always found as the time limit was reached. Tables 3 and 4 display some parameters for $\mathcal{F}_1$ and $\mathcal{FR}_2$ and experiments for each family of inequalities in an independent manner, as well as a combination of all of them. In fact, for $2{-}\text{part}_{xy}$, cycle, upper bound and the combination of all families of valid inequalities: the best objective function value, the optimality gap, the number of Branch & Cut nodes and the total number of added cuts, are presented. The subgraph separation routine produced a reduced number of violated inequalities, then this family of inequalities is not reported in the tables. Observing the results of Tables 3 and 4, one can conclude that the 2-partition inequalities provide the best reduction in the optimality gap, in comparison with the results shown when cycle and upper bound are included independently, and also in the case when all cuts are combined together. The latter is true despite of the fact that in the experiments, with the inclusion of all cuts, the number of nodes of the B&C approach is reduced significantly. This reduction in the number of nodes turns out even more evident if it is compared with the first set of experiments (Tables 1 and 2) when the Gurobi solver is used without the separation routines.

6 Conclusions

In this paper, a multi-weight graph partitioning problem is defined. The problem consists in partitioning a general graph in a fixed number of subsets of nodes such that multiple node weight constraints over each partition must be satisfied. The objective aims to minimize the total cost of edges with end-nodes in the same subset. The problem, because it generalizes other partitioning problems, has a theoretical interest for its study.

In order to solve the problem, two integer programs are proposed and several families of valid inequalities are proved. Three of them have proven to be effective for reducing significantly the number of B&B nodes and the optimality gap, when those are used as cuts. This effectiveness is confirmed with computational experiments on several instances of a real-world problem which arises in the context of a vehicle and pollster routing problem tackled by the authors in other work.

References

1. A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. Recent advances in graph partitioning. In Lasse Kliemann and Peter Sanders, editors, *Algorithm Engineering: Selected Results and Surveys*, pages 117–158, Cham, 2016. Springer International Publishing.
2. R. C. Carlson and G. L. Nemhauser. Scheduling to minimize interaction cost. *Operations Research*, 14(1):52–58, 1966.
3. Sunil Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, March 1993.
4. Nicos Christofides and P. Brooker. The optimal partitioning of graphs. *SIAM Journal on Applied Mathematics*, 30(1):55–69, 1976.
5. Jamie Fairbrother, Adam N. Letchford, and Keith Briggs. A two-level graph partitioning problem arising in mobile wireless communications. *Computational Optimization and Applications*, 69(3):653–676, November 2017.
6. Neng Fan and Panos M. Pardalos. Linear and quadratic programming approaches for the general graph partitioning problem. *Journal of Global Optimization*, 48(1):57–71, January 2010.
7. C. Ferreira, A. Martin, C. de Souza, R. Weismantel, and L. Wolsey. The node capacitated graph partitioning problem: A computational study. *Math. Program.*, 81:229–256, 1998.

8. M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Math. Program.*, 45:59–96, 1989.
9. Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2018.
10. Sandra Gutierrez, Andres Miniguano, Diego Recalde, Luis M. Torres, Ramiro Torres, and Pablo Zuleta. Integrated Vehicle and Pollster Routing. Technical report, Escuela Politécnica Nacional, Department of Mathematics, 07 2019.
11. B. Hendrickson and T. G. Kolda. Graph partitioning models for parallel computing. *Parallel Computing*, 26(12):1519–1534, 2000.
12. A. Kahng, J. Lienig, I. Markov, and J. Hu. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer Publishing Company, Incorporated, 1st edition, 2011.
13. B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, Feb 1970.
14. M. Labbé and F. A. Özsoy. Size-constrained graph partitioning polytopes. *Discr. Math.*, 310:3473–3493, 2010.
15. J. E. Mitchell. Branch-and-cut for the k-way equipartition problem. *Technical report, Department of Mathematical Sciences, Rensselaer Polytechnic Institute*, 2001.
16. John E. Mitchell. Realignment in the national football league: Did they do it right? *Naval Research Logistics (NRL)*, 50(7):683–701, 2003.
17. Diego Recalde, Daniel Severín, Ramiro Torres, and Polo Vaca. An exact approach for the balanced k-way partitioning problem with weight constraints and its application to sports team realignment. *Journal of Combinatorial Optimization*, 36(3):916–936, February 2018.

	2-part$_{xy}$				cycle				upper bound				All cuts together					
	Obj	Gap	Cuts	Nodes	Obj	Gap	Cuts	Nodes	Obj	Gap	Cuts	Nodes	Obj	Gap	2-part$_{xy}$	Cycles	UB	Nodes
	3324.67	0.00	1228	3097	3324.67	0.00	0	19619	3324.67	16.13	4	13809	3324.67	0.00	1326	0	4	3774
	2242.02	0.00	1917	11328	2242.02	0.00	5002	45578	2263.19	41.34	26	9545	2242.02	0.00	2209	5001	58	11496
(30,4)	3081.71	0.00	3693	16494	3096.42	23.29	0	41790	3096.42	46.78	10	10426	3096.42	7.29	3518	0	11	11768
	2533.44	0.00	1775	3082	2440.49	0.00	0	9693	2440.49	0.00	0	9693	2533.44	0.00	1775	0	0	3082
	1909.96	0.00	1510	5909	1909.96	0.00	5001	23724	1915.09	20.54	91	12363	1909.96	0.00	1482	5002	22	6001
	3887.53	50.74	5003	11087	4059.84	83.12	0	5998	4791.22	89.79	0	1732	4015.44	64.39	5003	0	0	2914
	3020.84	27.05	4921	10109	3811.78	82.26	0	6170	4785.39	90.87	12	1780	3020.84	44.32	3716	0	6	3182
(40,5)	3342.41	27.57	5001	11040	3342.41	28.63	5003	775289	4411.25	92.74	63	1525	3482.18	41.20	4385	5001	20	3856
	4311.32	22.09	5002	12216	4311.32	27.79	5002	739929	4934.65	90.80	0	1722	4381.35	28.28	5001	5002	159	7323
	2666.49	28.66	5001	9602	3018.39	70.56	0	6133	3365.36	85.11	2	1802	2666.49	38.80	4837	2	0	3591
	4711.79	57.67	5003	8279	5683.00	91.25	0	2049	5690.88	91.93	108	575	5128.35	75.56	5003	0	17	1549
	4733.61	64.68	5004	10518	6250.06	90.40	0	1986	7850.24	93.33	0	370	6430.51	82.23	5004	0	0	1692
(50,6)	4176.53	46.26	5002	12633	5930.39	86.74	0	1968	6246.73	92.95	0	513	4460.44	60.37	5002	0	32	1658
	5332.54	57.75	5001	8289	5537.26	61.78	5001	220097	10101.30	96.42	0	504	5967.49	69.60	5003	5001	405	2999
	4101.24	50.02	5001	9937	5775.03	89.44	0	2076	6207.62	90.17	0	580	4147.20	60.59	5003	0	4	1783
	4913.86	63.64	5002	6697	6229.89	91.78	0	1353	8426.52	94.47	0	284	5930.16	80.48	5002	0	0	1525
	3883.90	61.72	5004	5233	4581.18	75.44	5002	102753	6316.71	96.21	1	393	5066.65	76.31	5004	1911	0	1536
(55,6)	5372.43	68.51	5004	7950	6150.16	82.47	5003	107386	11073.80	97.17	0	281	6517.47	83.48	5004	5002	38	1656
	4878.72	58.22	5001	7013	4914.08	75.48	5002	98541	10140.60	97.06	0	295	4878.72	74.67	5001	3202	7	1459
	4670.29	62.04	5003	9764	7373.84	91.93	0	1024	7835.11	91.01	2	275	7181.84	80.79	5003	0	167	1527

Table 3 Solving $\mathcal{F}_1$ using valid inequalities.

	2-part$_x$				cycle				upper bound				All cuts together					
	Obj	Gap	Cuts	Nodes	Obj	Gap	Cuts	Nodes	Obj	Gap	Cuts	Nodes	Obj	Gap	2−part$_x$	Cycles	UB	Nodes
	3324.67	0.00	1193	2481	3324.67	0.00	0	18773	3324.67	20.05	1	13509	3324.67	0.00	1193	0	0	2481
	2242.02	0.00	1921	11195	2242.02	0.00	5003	36967	2271.98	39.06	36	9698	2242.02	0.00	1883	5002	13	8572
(30,4)	3081.71	0.00	2911	17373	3096.42	21.36	0	43453	3081.71	38.78	89	10438	3081.71	13.50	2721	0	0	9727
	2566.72	0.00	1770	3450	2440.49	0.00	0	13259	2440.49	0.00	0	13259	2566.72	0.00	1770	0	0	3450
	1909.96	0.00	1689	9286	1909.96	0.00	5001	28364	1909.96	24.56	6	12051	1909.96	0.00	1706	5001	5	8973
	3752.90	52.44	5001	5120	4554.03	84.00	0	8709	4751.16	91.15	0	1996	3752.90	58.61	4318	0	0	3079
	3020.84	33.22	4484	7000	3757.13	79.57	0	8163	4311.81	89.38	0	2155	3020.84	43.11	3736	0	0	3335
(40,5)	3342.41	27.21	4068	8589	3451.75	37.67	5001	372252	3668.84	88.05	16	2230	3342.41	37.22	2929	5002	22	3183
	4392.06	29.97	4849	7737	4311.32	33.54	5003	287378	5731.82	90.90	4	1866	4381.32	37.84	3773	5003	1	3671
	2661.67	32.79	5001	6978	2988.85	68.47	0	6034	2988.85	79.25	1	2068	2666.49	39.16	4535	0	0	3713
	4631.28	68.56	4437	1757	5127.11	90.30	0	1939	6823.78	94.21	102	536	4641.85	67.56	4135	0	1	1521
	5794.39	78.54	5001	2037	5914.10	89.48	0	1923	6288.01	90.45	0	502	5794.39	78.54	4896	0	0	1660
(50,6)	4926.45	63.16	4632	1771	5492.48	85.28	0	1910	6857.41	93.58	2	517	5696.60	69.01	3876	0	0	1364
	5850.26	70.79	4562	1935	5257.26	65.60	5002	108094	8373.99	95.68	0	372	6169.55	72.30	4100	1835	0	1596
	4132.87	59.53	4743	1842	5869.67	88.74	0	2021	6102.18	90.76	0	583	4132.87	59.61	4229	0	0	1518
	6308.75	79.86	4286	1374	7259.43	93.89	0	1167	10057.10	94.87	0	269	6308.75	79.86	3916	0	0	1256
	4489.25	71.28	4013	1190	3976.48	75.99	5002	49412	6044.77	93.63	0	255	4489.25	71.28	3970	432	0	1167
(55,6)	4484.42	75.95	4260	1444	4822.35	79.75	5002	50750	8901.15	96.73	0	237	4552.32	76.31	4013	466	2	1348
	5898.74	71.20	4386	1352	5273.38	83.00	5003	34518	8417.85	96.46	0	274	5898.74	71.20	4171	311	0	1278
	5410.17	73.34	4807	1540	6916.64	91.40	0	1197	7750.09	91.55	0	259	5445.44	73.51	4679	0	0	1480

Table 4 Solving $\mathcal{FR}_2$ using valid inequalities.