Computers & Industrial Engineering 135 (2019) 593–600

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Mathematical formulations for the K clusters with fixed cardinality problem

G.M. Gonçalves, L.L. Lourenço*

Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, P 2829-516 Monte da Caparica, Portugal
Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, P 2829-516 Monte da Caparica, Portugal

ARTICLE INFO

Keywords:
Combinatorial optimization
Clusters with fixed cardinality
Mixed integer linear formulations
Valid inequalities
Upper bounds

ABSTRACT

In this paper we propose some mixed integer linear programming formulations for the K clusters with fixed cardinality problem. These formulations are strengthened by valid inequalities and all the mixed integer linear models are compared from a theoretical and practical point of view. The continuous linear relaxation bounds of the developed models are tested on randomly generated instances, by using standard software, with promising results.

1. Introduction

The K clusters with fixed cardinality problem (KCCP) is to compute K disjoint clusters, each one with exactly M_k items, selected from a set of N items ($\sum_{k=1}^{N} M_k < N$), maximizing the total similarity among the items in the same cluster. The similarity between each pair of items (i, j) is a nonnegative value s_{ij} $(0 \leqslant s_{ij} \leqslant 1)$.

The KCCP is a NP-hard combinatorial optimization problem. In fact, it has the classic *k-cluster* problem as a particular case, which is NP-hard, see Billionnet (2005). To prove the computational complexity suppose that there is only 1 cluster with cardinality $M_1 < N$.

The KCCP was first introduced by Gonçalves and Lourenço (2009), where a mixed integer linear programming (MILP) formulation for the problem was proposed, as well as, a strengthened reformulation. In that work, the empirical experiments were only performed for 10 small instances with 13 items. For bigger instances, no optimum values were obtained. It is therefore crucial to study different formulations for the KCCP.

KCCP is a clustering type problem and related clustering type problems, however with different objectives or constraints, are described in Bruglieri, Ehrgott, Hamacher, and Maffioli (2006). Also a survey of mathematical programming models for clustering problems can be found in Hansen and Jaumard (1997).

Applications of the KCCP problem include software design for web search, customer segmentation, marketing area, document categorization, and scientific data analysis. For instance, when making a product search in the web, groups of other products are proposed in advertising windows to the user, based on previous searches. These products are grouped by similarity into groups with different sizes. Another application comes up in large supermarkets to select different products to be placed together, in view of a specific marketing strategy (Cavique, 2004). In this case, each cluster has a fixed number of products, where the similarity between each pair of products is based on the frequency that they are simultaneously bought. Still another important application of this problem arises in the financial area, to group N customers in K portfolios with M_k customers each one, maximizing the similarity between them, based on their profiles.

The paper is organized as follows: in Section 2 we present several mathematical formulations for the KCCP and in Section 3 the computational experiments performed in order to evaluate the proposed models are reported. The paper ends with some conclusions drawn from the work undertaken and some directions for future research.

2. Formulations

In order to formulate the KCCP consider the following notation:

i, j – items indexes $(i, j \in \{1, \dots, N\})$,
k – cluster index $(k \in \{1, \dots, K\})$,
N – number of items $(N \in \mathbb{N})$,
K – number of clusters $(K \in \mathbb{N},\ K < N)$,
M_k – number of items per cluster $k (M_k \in \mathbb{N},\ \sum_k M_k < N)$,
s_{ij} – similarity between items i and j, element of a symmetric matrix with diagonal elements equal to zero $(0 \leqslant s_{ij} \leqslant 1)$.

Next, the decision variables to assign the items to the clusters are defined. Let x_{ik} be a binary variable indicating whether item i is in cluster k $(=1)$ or not $(=0)$, $(i = 1, \dots, N;\ k = 1, \dots, K)$ and let also y_{ijk} be

* Corresponding author at: Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, P 2829-516 Monte da Caparica, Portugal.

E-mail addresses: gmsg@fct.unl.pt (G.M. Gonçalves), lll@fct.unl.pt (L.L. Lourenço).

https://doi.org/10.1016/j.cie.2019.06.028
Received 9 October 2018; Accepted 13 June 2019
Available online 19 June 2019
0360-8352/

a binary variable indicating whether items i and j are in the same cluster k (=1) or not (=0) ($i = 1, ..., N-1$; $j = i+1, ..., N$; $k = 1...,K$).

The KCCP may be formulated as the following quadratic problem:

$$(\text{Q})\ max \sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}x_{ik}x_{jk} \tag{1}$$

$$s.\ t.: \sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1, ..., N \tag{2}$$

$$\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1, ..., K \tag{3}$$

$$x_{ik} \in \{0, 1\} \quad i = 1, ..., N;\ k = 1, ..., K. \tag{4}$$

The objective function (1) gives the total similarity, which is the sum of the similarity values between pairs of items in the same cluster. The set of constraints (2) forces each item to belong to one cluster at most. Cardinality constraints (3) do not allow violation of the number of items in each cluster.

In the sequel, some MILP formulations for the KCCP are presented, obtained by linearizing the objective function (1) of the previous model.

Note that, the concave envelope for the bilinear terms $x_{ik}x_{jk}$ over the domain $(x_{ik}, x_{jk}) \in [0, 1] \times [0, 1]$, for each i, j, k, is obtained by introducing the variables y_{ijk} which replaces every occurrence of the product $x_{ik}x_{jk}$ in the problem Q (McCormick, 1976) and satisfy the following relationships $y_{ijk} \leqslant x_{ik}$ and $y_{ijk} \leqslant x_{jk}$. According with the parameters and the variables definition above, we obtain the following MILP problem:

$$\left(\text{F1}\right) \max \sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}y_{ijk} \tag{5}$$

$$s.\ t.: y_{ijk} \leqslant x_{ik} \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K \tag{6}$$

$$y_{ijk} \leqslant x_{jk} \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K \tag{7}$$

$$\begin{aligned}
&\sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1, ..., N\\
&\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1, ..., K\\
&x_{ik} \in \{0, 1\} \quad i = 1, ..., N;\ k = 1, ..., K\\
&0 \leqslant y_{ijk} \leqslant 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K.
\end{aligned} \tag{8}$$

This corresponds to the classical formulation due to Glover and Woolsey (1974). Constraints (6) and (7) result in $y_{ijk} = 1$ whenever $x_{ik} = x_{jk} = 1$, because the objective is to maximize, and result in $y_{ijk} = 0$ otherwise.

By considering now the convex envelope of the bilinear terms $x_{ik}x_{jk}$ over the domain $(x_{ik}, x_{jk}) \in [0, 1] \times [0, 1]$, for each i, j, k, we get the constraints $y_{ijk} \geqslant x_{ik} + x_{jk} - 1$. Including this constraint in the model, instead of (6) and (7), once the problem is of maximization, all the variables y_{ijk} will be equal to 1. It is then necessary to include a constraint in the model which forces $y_{ijk} = 0$ if $x_{ik} = 0$ or $x_{jk} = 0$. This is achieved with the star equalities $\sum_{i=1, i\neq j}^{N} y_{ijk} = \left(M_k - 1\right)x_{jk}$. Then, an alternate MILP formulation for the KCCP follows:

$$\left(\text{F2}\right) \max \sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}y_{ijk}$$

$$s.\ t.: y_{ijk} \geqslant x_{ik} + x_{jk} - 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K \tag{9}$$

$$\begin{aligned}
&\sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1, ..., N\\
&\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1, ..., K\\
&\sum_{i=1}^{j-1} y_{ijk} + \sum_{i=j+1}^{N} y_{jik} = \left(M_k - 1\right)x_{jk} \quad j = 1, ..., N;\ k = 1, ..., K
\end{aligned} \tag{10}$$

$$\begin{aligned}
&x_{ik} \in \{0, 1\} \quad i = 1, ..., N;\ k = 1, ..., K\\
&0 \leqslant y_{ijk} \leqslant 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K.
\end{aligned}$$

Equations (10) force $y_{ijk} = 1$ for $M_k - 1$ variables y_{ijk}, for fixed j and k, and constraints (9) force $y_{ijk} = 1$ whenever $x_{ik} = 1 = x_{jk}$.

In the following model, constraints (10) are replaced by one constraint defining the total number of variables y_{ijk} which are equal to 1.

$$\left(\text{F3}\right) \max \sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}y_{ijk}$$

$$\begin{aligned}
&s.\ t.: y_{ijk} \geqslant x_{ik} + x_{jk} - 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K\\
&\sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1, ..., N\\
&\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1, ..., K\\
&\sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} y_{ijk} = \sum_{k=1}^{K} \frac{M_k!}{2!\,(M_k-2)!}
\end{aligned} \tag{11}$$

$$\begin{aligned}
&x_{ik} \in \{0, 1\} \quad i = 1, ..., N;\ k = 1, ..., K\\
&0 \leqslant y_{ijk} \leqslant 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1, ..., K.
\end{aligned}$$

Note that Eq. (11) results by adding constraints (10) for all j and k values. This is a more compact formulation than the previous one.

A new formulation based on the linearization proposed by Glover (1975) can be defined for the KCCP. Observe that the objective function of the quadratic formulation Q, previously presented, can be rewritten in the following way:

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}x_{ik}x_{jk} &= \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\left(x_{ik}\sum_{j=i+1}^{N} s_{ij}x_{jk} + x_{ik}\sum_{j=1}^{i-1} s_{ji}x_{jk}\right)\\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} x_{ik}l_{ik}(x),
\end{aligned} \tag{12}$$

where

$$l_{ik}(x) = \sum_{j=i+1}^{N} s_{ij}x_{jk} + \sum_{j=1}^{i-1} s_{ji}x_{jk}.$$

Let us define the set $S_i = \{s_{ij}:\ i \neq j,\ j = 1, ..., N\}$, for $i = 1, ..., N$ and the parameters equal to the sum of the M_k biggest values of the set S_i, for $i = 1, ..., N,\ \ k = 1, ..., K$.

The concave envelope for the bilinear terms $x_{ik}l_{ik}(x)$ over the domain $(x_{ik}, l_{ik}(x)) \in [0, 1] \times [0, UG_{ik}]$, for each i, k, is obtained by introducing the variables z_{ik} which replaces every occurrence of the product $x_{ik}l_{ik}(x)$ in the problem Q (McCormick, 1976) and satisfies the following relationships $z_{ik} \leqslant l_{ik}(x)$ and $z_{ik} \leqslant UG_{ik}x_{ik}$.

The new continuous variables z_{ik} are then defined as $z_{ik} = l_{ik}(x)$, if $x_{ik} = 1$ and $z_{ik} = 0$, if $x_{ik} = 0$, for $i = 1, ..., N,\ \ k = 1, ..., K$ and the formulation is the following:

$$\left(\text{F4}\right) \max \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} z_{ik} \tag{13}$$

$$s.\ t.: \sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1,\ldots,N$$

$$\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1,\ldots,K$$

$$z_{ik} \leqslant l_{ik}(x) \quad i = 1,\ldots,N;\ k = 1,\ldots,K \tag{14}$$

$$z_{ik} \leqslant UG_{ik} x_{ik} \quad i = 1,\ \ldots,N;\ k = 1,\ \ldots,K \tag{15}$$

$$x_{ik} \in \{0,\ 1\} \quad i = 1,\ldots,N;\ k = 1,\ldots,K$$

$$z_{ik} \geqslant 0 \quad i = 1,\ldots,N;\ k = 1,\ldots,K. \tag{16}$$

Based on a formulation presented by Billionnet (2005) for the heaviest k-subgraph problem, we build the next model. Consider the parameters LB_{ik} equal to the sum of the $M_k - 1$ lowest values of the set S_i, for $i = 1,\ \ldots,N,\ \ k = 1,\ \ldots,K$, and UB_{ik} equal to the sum of the $M_k - 1$ biggest values of the set S_i, for $i = 1,\ \ldots,N,\ \ k = 1,\ \ldots,K$.

Define also the new continuous variables $t_{ik} = l_{ik} - LB_{ik}$ if $x_{ik} = 1$ and $t_{ik} = 0$ if $x_{ik} = 0$ for $i = 1,\ \ldots,N,\ \ k = 1,\ \ldots,K$.

The formulation is then the following:

$$\left(\text{F5}\right) \max \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} LB_{ik} x_{ik} + \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik} \tag{17}$$

$$s.\ t.: \sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1,\ldots,N$$

$$\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1,\ldots,K$$

$$t_{ik} \leqslant l_{ik}(x) - LB_{ik} \quad i = 1,\ldots,N;\ k = 1,\ldots,K \tag{18}$$

$$t_{ik} \leqslant (UB_{ik} - LB_{ik}) x_{ik} \quad i = 1,\ \ldots,N;\ k = 1,\ \ldots,K \tag{19}$$

$$t_{ik} \geqslant 0 \quad i = 1,\ \ldots,N;\ k = 1,\ \ldots,K \tag{20}$$

$$x_{ik} \in \{0,\ 1\} \quad i = 1,\ \ldots,N;\ k = 1,\ \ldots,K.$$

Note that if $x_{ik} = 0$ then, from (19), $t_{ik} = 0$ and in this case we have a zero at the objective function. Otherwise, if $x_{ik} = 1$ then, from (18) and (19), it follows $t_{ik} \leqslant l_{ik}(x) - LB_{ik}$. Once the problem is of maximization type, we have, at the optimum, $t_{ik} = l_{ik}(x) - LB_{ik}$. By substituting t_{ik} at the objective function it results in $\frac{1}{2} l_{ik}(x)$, which is the solution's cost.

The five MILP formulations for the KCCP, previously presented, are next compared from a theoretical point of view.

By denoting $\overline{P}$ the continuous linear relaxation of problem P and by $v(\overline{P})$ the optimum value of $\overline{P}$, Billionnet (2005) proved that, for $K = 1$ (corresponding to the heaviest k-subgraph problem), $v(\overline{F5}) \leqslant v(\overline{F4'})$ and $v(\overline{F1}) \leqslant v(\overline{F4'})$, where $F4'$ is the Glover formulation considering the parameter UG'_{ik} equal to the sum of all the M_k values of the set S_i, instead of . These results are straightforward generalized for the KCCP with $K > 1$, in the following two propositions. Observe that model $F4'$ is equal to model $F4$ except for the constraints (15), where is replaced by UG'_{ik}. Denote the new constraints by (15'). Obviously, $F4$ is a strengthened version of $F4'$.

Proposition 1. *$v(\overline{F5}) \leqslant v(\overline{F4})$ for the KCCP with $1 \leqslant K < N$.*

Proof. Let $(\overline{x},\ \overline{t})$ be a feasible solution of $\overline{F5}$. Its objective function value is

$$\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} LB_{ik}\overline{x}_{ik} + \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} \overline{t}_{ik}.$$

Let now $(\overline{x},\ \overline{z})$ be a feasible solution of $\overline{F4}$, such that $\overline{z}_{ik} = \overline{t}_{ik} + LB_{ik}\overline{x}_{ik}$, $\forall\ i = 1,\ \ldots,N,\ \ \forall\ k = 1,\ \ldots,K$.

From the constraints (18) we obtain

$$\overline{t}_{ik} \leqslant l_{ik}(\overline{x}) - LB_{ik} \Rightarrow \overline{t}_{ik} + LB_{ik}\overline{x}_{ik} \leqslant l_{ik}(\overline{x}) \Leftrightarrow \overline{z}_{ik} \leqslant l_{ik}(\overline{x}),$$

$\forall\ i = 1,\ \ldots,N\ ;\ \forall\ k = 1,\ \ldots,K$, because $0 \leqslant \overline{x}_{ik} \leqslant 1$ and $LB_{ik} \geqslant 0,\ \ \forall\ i$ and $\forall\ k$. The constraints (14) are satisfied. On the other hand, from (19), we get

$$\overline{t}_{ik} \leqslant (UB_{ik} - LB_{ik})\overline{x}_{ik} \Rightarrow \overline{t}_{ik} + LB_{ik}\overline{x}_{ik} \leqslant UB_{ik}\overline{x}_{ik},$$

$\forall\ i = 1,\ \ldots,N,\ \ \forall\ k = 1,\ \ldots,K$. As, by definition, $UB_{ik} \leqslant UG_{ik}$, then

$$\overline{t}_{ik} + LB_{ik}\overline{x}_{ik} \leqslant UG_{ik}\overline{x}_{ik} \Leftrightarrow \overline{z}_{ik} \leqslant UG_{ik}\overline{x}_{ik},$$

i.e., the constraints (15) are satisfied. Then this solution is feasible to model $\overline{F4}$.

The value of the objective function of this solution is

$$\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} \overline{z}_{ik} = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\left(\overline{t}_{ik} + LB_{ik}\overline{x}_{ik}\right),$$

which is equal to the objective function value of model $\overline{F5}$. □

Proposition 2. *$v(\overline{F1}) \leqslant v(\overline{F4'})$ for the KCCP with $1 \leqslant K < N$.*

Proof. Let $(\overline{x},\ \overline{y})$ be a feasible solution of $\overline{F1}$. Its objective function value is

$$\sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}\overline{y}_{ijk}.$$

Let $(\overline{x},\ \overline{z})$ be a feasible solution of $\overline{F4'}$, such that $\overline{z}_{ik} = \sum_{j=i+1}^{N} s_{ij}\overline{y}_{ijk} + \sum_{j=1}^{i-1} s_{ji}\overline{y}_{jik}$.

From the constraints (6) and (7) we obtain

$$\overline{z}_{ik} \leqslant \sum_{j=i+1}^{N} s_{ij}\overline{x}_{jk} + \sum_{j=1}^{i-1} s_{ji}\overline{x}_{jk} \Leftrightarrow \overline{z}_{ik} \leqslant l_{ik}(\overline{x}),$$

which are the constraints (14). On the other hand, from the constraints (6) and (7), we get

$$\overline{z}_{ik} \leqslant \sum_{j=i+1}^{N} s_{ij}\overline{x}_{ik} + \sum_{j=1}^{i-1} s_{ji}\overline{x}_{ik} \Leftrightarrow \overline{z}_{ik} \leqslant \left(\sum_{j=i+1}^{N} s_{ij} + \sum_{j=1}^{i-1} s_{ji}\right)\overline{x}_{ik} \leqslant UG'_{ik}\overline{x}_{ik}$$

Consequently $\overline{z}_{ik} \leqslant UG'_{ik}\overline{x}_{ik}$, which are the constraints (15'). Then this solution is feasible to model $\overline{F4'}$.

The value of the objective function of this solution is

$$\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} \overline{z}_{ik} = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\left(\sum_{j=i+1}^{N} s_{ij}\overline{y}_{ijk} + \sum_{j=1}^{i-1} s_{ji}\overline{y}_{jik}\right) = \sum_{k=1}^{K}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij}\overline{y}_{ijk}$$

which is equal to the objective function of model $\overline{F1}$. □

Note that model F_4 is not comparable to model $F1$, in the sense that for some instances, $\overline{F4}$ yields a tighter upper bound than the one given by $\overline{F1}$, while for other instances, the opposite result is yield. This can be observed in the computational experiments, in Section 3.2.

The models $F2$ and $F3$ are theoretically comparable. The constraints (11) of model $F3$ result from the sum of constraints (10) of model $F2$, for all j and all k. It follows that,

Proposition 3. *$v(\overline{F2}) \leqslant v(\overline{F3})$ for the KCCP with $1 \leqslant K < N$.*

It is possible to strengthen model $F1$ by including in the same formulation the inequalities (9) and (10). The resulting model is

$$\left(\mathrm{F6}\right) \max \sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} y_{ijk}$$

$$s.\ t.:\ y_{ijk} \geqslant x_{ik} + x_{jk} - 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1,...,K \quad (9)$$

$$y_{ijk} \leqslant x_{ik} \quad 1 \leqslant i < j \leqslant N;\ k = 1,...,K \quad (6)$$

$$y_{ijk} \leqslant x_{jk} \quad 1 \leqslant i < j \leqslant N;\ k = 1,...,K \quad (7)$$

$$\sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1,...,N \quad (2)$$

$$\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1,...,K \quad (3)$$

$$\sum_{i=1}^{j-1} y_{ijk} + \sum_{i=j+1}^{N} y_{jik} = \left(M_k - 1\right) x_{jk} \quad j = 1,...,N;\ k = 1,...,K \quad (10)$$

$$x_{ik} \in \{0, 1\} \quad i = 1,...,N;\ k = 1,...,K \quad (4)$$

$$0 \leqslant y_{ijk} \leqslant 1 \quad 1 \leqslant i < j \leqslant N;\ k = 1,...,K. \quad (8)$$

Proposition 4. $v(\overline{F6}) \leqslant v(\overline{F1})$ *for the KCCP with* $1 \leqslant K < N$.

Proof. The formulation $F6$ is based on the formulation $F1$ and it includes additionally the constraints (9) and (10), therefore $\overline{F6}$ is stronger than $\overline{F1}$. □

Proposition 5. $v(\overline{F6}) \leqslant v(\overline{F2})$ *for the KCCP with* $1 \leqslant K < N$.

Proof. The formulation $F6$ is based on the formulation $F2$ and it includes additionally the constraints (6) and (7), therefore $\overline{F6}$ is stronger than $\overline{F2}$. □

It is also possible to strengthen the model F3 by including in the same model the inequalities (10), resulting in the problem

$$\left(\mathrm{F7}\right) \max \sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} y_{ijk}$$

$$s.\ t.:\ y_{ijk} \geqslant x_{ik} + x_{jk} - 1 \quad 1 \leqslant i < j \leqslant N; k = 1,...,K \quad (9)$$

$$\sum_{k=1}^{K} x_{ik} \leqslant 1 \quad i = 1,...,N \quad (2)$$

$$\sum_{i=1}^{N} x_{ik} = M_k \quad k = 1,...,K \quad (3)$$

$$\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} y_{ijk} = \sum_{k=1}^{K} \frac{M_k!}{2!\,(M_k - 2)!} \quad (11)$$

$$\sum_{i=1}^{j-1} y_{ijk} + \sum_{i=j+1}^{N} y_{jik} = \left(M_k - 1\right) x_{jk} \quad j = 1,...,N; k = 1,...,K \quad (10)$$

$$x_{ik} \in \{0, 1\} \quad i = 1,...,N; k = 1,...,K \quad (4)$$

$$0 \leqslant y_{ijk} \leqslant 1 \quad 1 \leqslant i < j \leqslant N; k = 1,...,K. \quad (8)$$

Proposition 6. $v(\overline{F7}) \leqslant v(\overline{F3})$ *for the KCCP with* $1 \leqslant K < N$.

Proof. The formulation $F7$ is based on the formulation $F3$ and it includes additionally the constraints (10), therefore $\overline{F7}$ is stronger than $\overline{F3}$. □

Proposition 7. $v(\overline{F7}) \leqslant v(\overline{F2})$ *for the KCCP with* $1 \leqslant K < N$.

Proof. The formulation $F7$ is based on the formulation $F2$ and it includes additionally the constraints (11), therefore $\overline{F7}$ is stronger than $\overline{F2}$. □

Proposition 8. $v(\overline{F6}) \leqslant v(\overline{F7})$ *for the KCCP with* $1 \leqslant K < N$.

Proof. Let $(\bar{x}, \bar{y})$ be a feasible solution of $\overline{F6}$. Its objective function value is

$$\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \bar{y}_{ijk}.$$

Next it is proved that $(\bar{x}, \bar{y})$ is a feasible solution of $\overline{F7}$.

As the constraints (2)–(4), (8)–(10) are shared by the two models, it is only necessary to prove that Eq. (11) is satisfied.

The solution $(\bar{x}, \bar{y})$ satisfies the constraints (10), still satisfying (11), which is the sum of (10) for all i and for all j. The result follows because the value of the objective function of the corresponding solutions is the same for both models. □

In short the above results can be summarized in what follows:

(i) $v(\overline{F5}) \leqslant v(\overline{F4}) \leqslant v(\overline{F4'})$,
(ii) $v(\overline{F6}) \leqslant v(\overline{F1}) \leqslant v(\overline{F4'})$,
(iii) $v(\overline{F6}) \leqslant v(\overline{F7}) \leqslant v(\overline{F2}) \leqslant v(\overline{F3})$.

The next section presents computational experiments with all the previously models which are compared.

3. Computational experience

This section reports the computational experience performed with the MILP models and its continuous relaxations, proposed in the previous section for the KCCP. As no benchmark instances exist for the KCCP, a set of instances for this problem was generated, based on the instances for the k-cluster Problem, reported in the CEDRIC's Library of instances (http://cedric.cnam.fr/lamberta/Library/k-cluster.html) (Billionnet, 2005). The parameters used to generate the instances, as the generating process are presented in Section 3.1. The computational results for the generated instances and for the proposed models are shown in Section 3.2.

3.1. Test instances

The instances of the KCCP used in our computational experiments were generated on the basis of the instances for the k-cluster Problem, with $N = 40$, from CEDRIC's Library. Each one of the CEDRIC's instances is defined by a graph, by its density (d) and by the number M_1 of items in the cluster, which is equal to 10 $(\frac{1}{4}N)$, 20 $(\frac{1}{2}N)$ or 30 $(\frac{3}{4}N)$. There are three different graph densities, which are 0.25, 0.50 and 0.75. For each M_1 fixed and each density d fixed, there are 5 different graphs, with all edge weights equal to 1. The total number of different graphs is 15 and the set of 15 graphs considered for different M_1 values is always the same. Note that, for each graph, 3 different instances exist, each one with a different number M_1 of items in the cluster. The final number of instances is 45.

Recall that the KCCP aims to select M_k items to each cluster k ($k = 1, ...,K$) such that $\sum_{k=1}^{K} M_k < N$, maximizing the total similarity among the items. One instance of the KCCP is then characterized by the number N of items, by the number K of clusters, by the number M_k of items in the cluster k, for each k, and by the items' similarity values s_{ij}.

The KCCP test instances were obtained by considering $N = 40$ and they were based on the 15 graphs mentioned above. For each edge graph $[i, j]$, a positive weight s_{ij} was randomly generated strictly between 0 and 1, defining the similarity between items i and j. The remaining data of these instances were defined as follows. (see Table 1).

Instances with $K = 1, 2, 3$ and $\sum_{k=1}^{K} M_k = 10, 20, 30$ were generated. For $K = 1$ we generated 45 instances, 15 for each M_1 value. For $K = 2$ or $K = 3$, 90 instances were generated, 15 for each value of $\sum_{k=1}^{K} M_k$ and for each choice of M_k, according to Table 2. Two different options for

Table 1
CEDRIC's instance parameters.

K	M_1	Graph density (d)	N. Inst
1	10	(0.25, 0.50, 0.75)	15
	20	(0.25, 0.50, 0.75)	15
	30	(0.25, 0.50, 0.75)	15

Table 2
Instance parameters with $N = 40$.

K	$\sum_{k=1}^{K} M_k$	M_k			N. Inst
1	10	$M_1 = 10$			15
	20	$M_1 = 20$			15
	30	$M_1 = 30$			15
2	10	$M_1 = 5\left(\frac{1}{2}\right)$	$M_2 = 5\left(\frac{1}{2}\right)$		15
		$M_1 = 2\left(\frac{1}{5}\right)$	$M_2 = 8\left(\frac{4}{5}\right)$		15
	20	$M_1 = 10\left(\frac{1}{2}\right)$	$M_2 = 10\left(\frac{1}{2}\right)$		15
		$M_1 = 4\left(\frac{1}{5}\right)$	$M_2 = 16\left(\frac{4}{5}\right)$		15
	30	$M_1 = 15\left(\frac{1}{2}\right)$	$M_2 = 15\left(\frac{1}{2}\right)$		15
		$M_1 = 24\left(\frac{4}{5}\right)$	$M_2 = 6\left(\frac{1}{5}\right)$		15
3	10	$M_1 = 3\left(\frac{3}{10}\right)$	$M_2 = 3\left(\frac{3}{10}\right)$	$M_3 = 4\left(\frac{4}{10}\right)$	15
		$M_1 = 2\left(\frac{1}{5}\right)$	$M_2 = 3\left(\frac{3}{10}\right)$	$M_3 = 5\left(\frac{1}{2}\right)$	15
	20	$M_1 = 7\left(\frac{7}{20}\right)$	$M_2 = 7\left(\frac{7}{20}\right)$	$M_3 = 6\left(\frac{3}{10}\right)$	15
		$M_1 = 3\left(\frac{3}{20}\right)$	$M_2 = 7\left(\frac{7}{20}\right)$	$M_3 = 10\left(\frac{1}{2}\right)$	15
	30	$M_1 = 10\left(\frac{1}{3}\right)$	$M_2 = 10\left(\frac{1}{3}\right)$	$M_3 = 10\left(\frac{1}{3}\right)$	15
		$M_1 = 5\left(\frac{1}{6}\right)$	$M_2 = 10\left(\frac{1}{3}\right)$	$M_3 = 15\left(\frac{1}{2}\right)$	15

the M_k values were considered, one with balanced and another one with unbalanced values of M_k. As there are 6 different options for M_k, then there exists $6 \times 15 = 90$ instances.

Note that, for $K = 2,\ 3$, for d fixed and $\sum_k M_k$ also fixed, there are 2 different instances relative to the same graph. Then, for those d and $\sum_k M_k$ fixed, there are 10 instances because 5 different graphs exist for each density.

3.2. Computational results

All models were solved by using the standard mathematical software CPLEX. The algorithm provided by the ilog CPLEX 12.6, ran on a i7 computer with 3.60 GHz processor and 8 GB RAM. In all tests the following CPLEX parameters were considered: time limit = 7200 s, clocktype = 1, mip tol absmipgap = 0.0, mip tol mipgap = 0.0, mip tol integrality = 0.0, feasopt tolerance = 0, threads = 8, while the other standard CPLEX parameters were used. The computational tests were made for the instances described in the previous section.

Recall that for $K = 1$ and for each density there are 5 different graphs corresponding to 5 different instances. In Tables 3–5 presented below for $K = 1$, the first two columns are the graph density and the values of $\sum_k M_k$, respectively. The Gap, the cardinal of nodes and the CPU time presented are the average values for the 5 instances. The Gap at the root node of the search tree is equal to $\frac{v(\overline{P}) - v(P)}{v(P)} * 100\%$, where $v(P)$ is the optimum value of the problem and $v(\overline{P})$ is the linear relaxation optimum value of the same problem. When there is no optimal solution available, the best integer solution is considered for the gap computation.

For $K = 1$, in Tables 3–5, we note that the optimal solution was obtained for all models in the time limit of 7200 s, except for 1 instance.

From Table 3 the average gap for the set of instances corresponding to $d = 25$ and $\sum_k M_k = 10$, is 56.0 for $F1$ and 153.7 for $F2$, i.e., $F1$ is better than $F2$. However, from the same table, for $d = 50$ and $\sum_k M_k = 10$, the opposite result is observed. Then, those models are not comparable. Similar observations can be made when we compare the models $F1$ and $F3$, $F1$ and $F4$, $F1$ and $F5$, $F2$ and $F4$, $F2$ and $F5$, $F3$ and $F4$ (see Tables 3 and 4). We observe that for this set of instances the model $F5$ gave better average gaps than $F3$.

As expected $F5$ gave better results than $F4$, as illustrated in Table 4.

For $K = 1$, from Tables 3–5, we emphasize that the model F6 gave the best average gaps for all instances and in general gave the least average number of nodes. The CPU time for the same model was low in general. One may conclude that the constraints (6) and (7) are effective, which is verified when comparing the average gaps obtained from models $F2$ and $F6$ (see Proposition 5). Those constraints are valid inequalities for the convex hull of the feasible region of model $F2$, leaving to stronger upper bounds for the optimum value.

In the sequel, in Tables 6–8, for K = 2, and in Tables 9–11, for K = 3, the first two columns are the density and the values of $\sum_k M_k$. The Gap, the cardinal of nodes and the CPU time presented are the average values for 10 instances. Observe also that for $K = 2$ and K = 3, the optimal solution was not obtained for all models, in the time limit of 7200 s. In the column Uns. Inst. the number of unsolved instances is presented.

The models $F6$ and $F7$ had a better performance and gave the best average gaps, however did not gave the optimal solution for 5 instances

Table 3
Results for models $F1$, $F2$ and $F3$ for K = 1.

d	$\sum_k M_k$	F1			F2			F3		
		Average gap (%)	Average # nodes	Average CPU time (s)	Average gap (%)	Average # nodes	Average CPU time (s)	Average gap (%)	Average # nodes	Average CPU time (s)
25	10	56.0	451.2	0.8	153.7	951.2	7.7	153.9	565986.8	332.2
25	20	21.0	0.0	0.4	124.8	689.6	16.4	136.9*	5849150.8	7200.5
25	30	5.3	0.0	0.2	18.2	30.6	2.7	29.1	50316.8	42.1
50	10	107.4	63123.0	68.3	78.3	1744.2	10.7	78.5	103551.8	72.2
50	20	39.7	14495.2	15.9	92.5	7944.2	95.0	102.7	4419446.8	5022.1
50	30	12.2	914.6	1.3	10.2	58.8	2.4	15.8	5926.2	5.7
75	10	148.8	1328139.2	1582.4	48.0	1388.4	10.8	48.1	51464.0	36.0
75	20	54.0	751915.8	1023.1	62.9	19681.8	200.4	68.2	1715998.6	2005.5
75	30	18.3	92541.0	131.4	6.6	117.0	2.6	10.2	5263.0	5.4

* 1 instance was not solved.

Table 4
Results for models $F4$ and $F5$ for K = 1.

d	$\sum_k M_k$	F4			F5		
		Average gap (%)	Average # nodes	Average CPU time (s)	Average gap (%)	Average # nodes	Average CPU time (s)
25	10	58.4	8214.6	5.8	57.3	7684.8	5.4
25	20	23.2	4620.2	4.0	23.2	4620.2	3.9
25	30	6.8	749.0	0.6	6.7	1368.8	1.0
50	10	63.3	240782.2	85.6	52.5	203858.0	60.5
50	20	39.8	194170.4	164.0	38.9	158816.0	156.8
50	30	12.6	6270.0	6.8	7.8	5538.0	3.2
75	10	48.0	983746.0	230.1	36.0	300730.0	65.0
75	20	41.3	4665912.8	1909.4	35.7	4803433.2	1987.1
75	30	18.1	19794.6	18.2	7.8	10942.2	5.3

Table 5
Results for models $F6$ and $F7$ for K = 1.

d	$\sum_k M_k$	F6			F7		
		Average gap (%)	Average # nodes	Average CPU time (s) time (s)	Average gap (%)	Average # nodes	Average CPU time (s) time (s)
25	10	**47.4**	443.8	6.3	153.7	1010.6	9.0
25	20	**20.9**	733.0	10.5	124.8	1130.4	23.5
25	30	**2.4**	28.0	0.7	18.2	28.4	3.3
50	10	**40.9**	904.6	9.9	78.3	2092.4	12.7
50	20	**35.9**	6871.0	67.9	92.5	7353.4	98.9
50	30	**1.8**	37.2	0.9	10.2	50.8	2.8
75	10	**27.7**	1013.6	11.0	48.0	1565.8	10.9
75	20	**31.9**	20838.4	201.8	62.9	21864.2	226.9
75	30	**2.2**	88.8	1.5	6.6	99.6	3.1

The bold values are the best average gaps for K = 1.

(for $\sum_k M_k = 30$) in the time limit of 7200 s. These two models gave the least average number of nodes and better CPU times, in general.

Observe that for the model F1 with $d = 75$ and $\sum_k M_k = 20,\ 30$, the CPLEX gave the message out of memory for 3 instances. This was the worst model for instances with high density. The models $F2$ and $F3$ solved the same number of instances than $F6$ and $F7$, but gave worst average gaps.

For the instances with K = 2, $F4$ and $F5$ did not solved to optimality 1 instance with low density in the time limit of 7200 s, neither some instances with medium and high density. However $F5$ gave lower average gaps and solved more instances than $F4$.

From Tables 9–11, for K = 3, we can note that F6 gave the best average gaps, but the model $F2$ solved a bigger number of instances with lower CPU time. The models $F1$ and $F3$ gave the message "out of memory" for some instances, according to Table 9. $F4$ and $F5$ did not solve a big number of instances.

From the computational experience we have noted that, for K = 2 and K = 3, the models solved a bigger number of unbalanced instances when compared with the balanced ones. We can observe that, for all instances, the model $F6$ gave the best average gaps. However, for $K = 3$, the model $F2$, which has less constraints than $F6$, solved more instances, though the upper bound at the root node was highest.

Note that KCCP's instances for K = 4 and K = 5 were tested but, in most cases, no integer solutions were obtained and they are not displayed in this paper.

4. Conclusion

In this paper we presented the K clusters with fixed cardinality problem. We have proposed several mathematical formulations for this problem: a quadratic model, as well as 5 MILP formulations and 2 strengthened formulations.

The MILP models were compared from a theoretical and practical point of view. The continuous relaxation bounds of the models were tested on randomly generated instances, by using the CPLEX software. We concluded that model $F6$ was the adequate model to solve the majority of instances, however for denser instances, the model $F2$ solved more problems.

From the computational perspective, the KCCP is complex as it is NP-hard. As previously noted, in Section 3.2, several instances were not solved by using the models presented above. In the near future we intend to develop other models for the KCCP, with less dimensions, in order to get solution for bigger instances. On the other hand, we intend to develop heuristic methods to get good feasible solutions for the KCCP.

Table 6
Results for models $F1$, $F2$ and $F3$ for K = 2.

d	$\sum_k M_k$	F1				F2				F3			
		Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	104.8	30683.9	50.8		93.5	708.3	5.3		93.5	1279.5	9.1	
25	20	54.7	30081.5	41.4		161.9	10341	208.9		161.9	11464.7	353.9	
25	30	36.3	13009.3	20.2		137.2	13068.2	632.5		137.2	15019.9	978.2	
50	10	209.6	1911661.4	5684.2	2	46.3	1094.7	7.1		46.3	2261.3	13.7	
50	20	99.8	2909530.5	5986.5	6	91.2	63007.4	1168.4		91.2	90799.7	2137	
50	30	61.0	1511528.1	3167.3	2	81.3	107621.7	3707.7	5	81.2	125618.8	3756.9	5
75	10	303.9	1260554.8	7202.9	10	28.9	785.8	6.3		28.9	1116.9	8.5	
75	20	127.5**	1853950.5	7201.9	10	57.3	68256.3	1361.6		57.3	82657.9	1753.1	
75	30	77.0*	1583068.4	7202.3	10	52.1	120574.1	3843.2	5	52.2	118135.8	3846.7	5

* 1 out of memory.
** 2 out of memory.

Table 7
Results for models $F4$ and $F5$ for K = 2.

d	$\sum_k M_k$	F4				F5			
		Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	72.8	137641.1	67.7		54.4	60969.8	21.9	
25	20	52.6	474666.2	974.7	1	50.0	492048.3	907.3	1
25	30	36.6	286282.9	862.0		35.6	291656.5	838.9	
50	10	60.2	3002064.6	704.5		36.0	204069	52.9	
50	20	63.4	7727698.5	6461.5	7	54.8	11711874.6	4694.8	5
50	30	50.7	2150372.4	4955.3	6	46.3	2499795.6	4674.8	5
75	10	46.8	3855874.3	883.7		23.2	67664.9	19.3	
75	20	50.1	15597513.2	7200.2	10	39.6	16603875.2	5828.3	7
75	30	45.4	9298636	7200.6	10	36.5	9195046.1	6794.4	8

Table 8
Results for models $F6$ and $F7$ for K = 2.

d	$\sum_k M_k$	F6				F7			
		Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	**40.5**	401.1	7.0		**40.5**	1279.5	9.0	
25	20	**43.2**	8206.7	217.3		**43.2**	11464.7	355.3	
25	30	**32.7**	7885.5	293.9		**32.7**	15019.9	979.0	
50	10	**27.9**	721.4	10.9		**27.9**	2261.3	13.5	
50	20	**44.7**	60506.7	1291.8		**44.7**	90801.5	2128.5	
50	30	38.0	154712.2	3715.7	5	38.0	126865	3755.4	5
75	10	**17.8**	451.6	7.8		**17.8**	1116.9	8.5	
75	20	**32.2**	61544	1397.0		**32.2**	82657.9	1752.3	
75	30	28.6	125657.3	4026.6	5	28.5	119521.2	3845.0	5

The bold values are the best average gaps for K = 2.

Table 9
Results for models F1, F2 and F3 for K = 3.

d	$\sum_k M_k$	F1				F2				F3			
		Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	162.8	631313.1	1336.3		37.7	247.9	4.0		38.8	16285.8	33.3	
25	20	88.3	1077552.3	1914.0		123.3	42771.5	761.1		132.0	3639971	7203.7	10
25	30	68.1	619914.5	1452.7	1	175.6	75206.3	3998.4	3	194.9*	1333650.222	7206.3	10
50	10	355.4	1053571.9	7203.4	10	16.8	128.2	2.6		17.2	4810.4	10.7	
50	20	174.8	1626118.7	7203.2	10	64.4	173491.8	3372.1	1	67.1	3660462.4	7204.5	10
50	30	119.9*	1487386.2	7203.4	10	98.8	153867.8	7202.4	10	101.4***	1135275	7208.4	10
75	10	536.1	723131.5	7205.7	10	11.6	140.4	2.5		11.9	5916.7	13.7	
75	20	239.8***	1052124.8	7206.2	10	39.3	111741.2	2411.1		40.6	3486090.6	7201.8	10
75	30	157.5	885499.0	7207.2	10	60.8	134110.5	7202.0	10	63.8**	1264851.25	7206.5	10

* 1 out of memory.
** 2 out of memory.
*** 4 out of memory.

Table 10
Results for models F4 and F5 for K = 3.

d	$\sum_k M_k$	F4				F5			
		Average gap (%)	Average # nodes # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	67.5	1053852.3	446.6		27.7	5844.4	2.7	
25	20	75.8	2280365.4	6044.0	6	64.8	2532217.7	4419.1	4
25	30	67.4	1305227.5	6471.1	8	65.0	1393398.1	6426.8	8
50	10	54.9	3148944.1	858.4		13.8	2813	1.3	
50	20	67.1	14542317.2	7200.5	10	48.2	19693554.9	7200.2	10
50	30	73.9	5529738.9	7201.0	10	62.5	6181241.5	7201.3	10
75	10	50.2	15010802.5	3494.5	2	9.7	4316.7	1.7	
75	20	49.8	20920809.6	7200.3	10	30.7	23629115.6	7013.9	9
75	30	54.6	9250448.4	7200.3	10	42.7	12477813.2	7200.2	10

Table 11
Results for models F6 and F7 for K = 3.

d	$\sum_k M_k$	F6				F7			
		Average gap (%)	Average # nodes # nodes	Average CPU time (s)	Uns. inst.	Average gap (%)	Average # nodes	Average CPU time (s)	Uns. inst.
25	10	**20.1**	198.8	5.6		37.7	358.7	4.5	
25	20	**50.6**	39444.4	1608.4		123.3	58290.2	1361.0	
25	30	**56.9**	64274.7	4686.3	3	175.8	72115.3	4209.7	3
50	10	**10.5**	102.8	3.6		16.8	155.3	3.1	
50	20	**38.5**	113861.3	4862.2	5	64.4	177082	4127.2	3
50	30	**54.3**	72346.2	7202.1	10	98.5	139648.6	7202.4	10
75	10	**7.1**	120.4	3.9		11.6	172.5	3.1	
75	20	**24.7**	67368.5	3347.9	2	39.3	105077.8	2726.3	1
75	30	**37.8**	62833.9	7201.7	10	61.2	129095.4	7202.2	10

The bold values are the best average gaps for K = 3.

Funding

This work is partially supported by the Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2019 (Centro de Matemática e Aplicaçaões).

References

Billionnet, A. (2005). Different formulations for solving the heaviest k-subgraph problem. *Information Systems and Operational Research, 43*(3), 171–186.

Bruglieri, M., Ehrgott, M., Hamacher, H., & Maffioli, F. (2006). An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics, 154*, 1344–1357.

Cavique, L. (2004). Graph-based strutures for the market baskets analysis. *Investigação Operacional, 24*, 233–246.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer programs. *Management Science, 22*, 455–460.

Glover, F., & Woolsey, E. (1974). Converting the 0–1 polynomial programming problem to a 0–1 linear program. *Operations Research, 22*, 180–182.

Gonçalves, G. M., & Lourenço, L. L. (2009). A strengthened mixed-interger linear formulation for the K clusters problem with fixed cardinality. *Recent advances in applied mathematics, proceedings of the 14th WSEAS international conference on applied mathematics* (pp. 229–232). .

Hansen, P., & Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming, 79*, 191–215.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming, 10*, 147–175.