Different formulations for solving the heaviest *k*-subgraph problem

Alain Billionnet
CEDRIC-IIE, 18 allée Jean Rostand, 91025 Evry cedex, France
Alain.Billionnet@iie.cnam.fr

Abstract. *We consider the heaviest k-subgraph problem, i.e. determine a block of k nodes of a weighted graph (of n nodes) such that the total edge weight within the subgraph induced by the block is maximized. We compare from a theoretical and practical point of view different mixed integer programming formulations of this problem. Computational experiments when the weight of each edge is equal to 1 are reported.*

Key words: Heaviest *k*-subgraph problem, mixed integer linear programming, upper bounds, experiments.

1. Introduction

Given an undirected graph $G=(V,E)$ with $V=\{v_1,...,v_n\}$ and non-negative edge weights $w_{i,j}$ on edges $[v_i,v_j]\in E$, *the heaviest k-subgraph problem* (HSP) consists in determining a subset $S\subset V$ of k nodes such that the total edge weight of the subgraph induced by *S* is maximized. HSP is also known under the name of *p-dispersion problem*, *k-cluster problem* and *dense k-subgraph problem* (DSP) when all the edge weights are equal to 1. A straightforward quadratic 0-1 formulation of HSP is given by Q

$$(Q)\quad \begin{cases} \text{Maximize } f(x)=\sum_{(i,j)\in T} w_{i,j}x_ix_j \\ \text{s.t.} \\ \sum_{i=1}^{n} x_i = k & (1) \\ x_i\in\{0,1\} \quad (i=1,...,n) & (2) \end{cases}$$

where $T = \{(i, j) \in \{1,...,n\}^2 : i < j, [v_i, v_j] \in E\}$. The binary variable x_i is equal to 1 if and only if vertex v_i is put in the *k*-subgraph. The problem is NP-difficult even for bipartite graphs with $w_{i,j} = 1$ for all $[v_i, v_j] \in E$ [Corneil and Perl, 1984]. Many approximation results are known for HSP [Asahiro et al., 1996], [Hassin et al., 1997], [Kortsarz and Peleg, 1993], [Srivastav and Wolf, 1997] but no approximation algorithm with fixed ratio-bound have been found to date and the question of knowing if such an algorithm exists is open. Concerning the practical resolution of HSP a few works have been published. Kincaid [Kincaid, 1992] presented heuristic methods based on simulated annealing and Tabu search but, to the best of our knowledge, no experimental results have been published about the exact solution of the problem. A slightly different problem is considered by Erkut [Erkut, 1990] : given a graph $G = (V, E)$ and non-negative edge weights $w_{i,j}$ on edges $[v_i, v_j] \in E$, determine a subset $S \subset V$ of k nodes such that the weight of the minimum weight edge appearing in the subgraph of *G* induced by *S* is maximized. A classical combinatorial optimization problem, more general than HSP, is the so-called quadratic 0-1 knapsack problem (QKP). This problem can be viewed as the following graph problem: given an undirected graph $G = (V, E)$, non-negative weights $w_{i,j}$ on edges $[v_i, v_j] \in E$ and p_i on nodes $v_i \in V$, determine a subset $S \subset V$ such that the total node weight of S is less than *k,* and the total edge weight of the subgraph induced by *S* is maximized. Several algorithms have been proposed for QKP (see, for example, [Hammer and Rader, 1997], [Billionnet et al. 1999] and [Caprara et al., 1999]) which allow instances with a few hundred nodes to be solved. However, it seems that HSP, which corresponds to QKP where all the node weights are equal, is much more difficult to solve in practice.

We address in this paper the exact solution of HSP. The aim of the paper is to show how HSP can be solved by using a classical approach: mixed integer programming. The obtained results eventually can be used to evaluate the efficiency of specific algorithms for HKP. A significant advantage of our approach is that it can be handled via a mixed integer programming tool. That considerably reduces the degree of difficulty of implementation relative to other approaches since only standard, commercially available, software is required. The technique is known and well tried but it is necessary to carefully implement it since some formulations may require a prohibitive computation time (see, for example, [Salkin and Mathur, 1989] and [Beale, 1988]). In Section 2 we propose four different

formulations of HSP as mixed integer linear programs. In Section 3, these formulations are theoretically compared from an upper bound point of view, the considered upper bounds being the optimal values of the continuous relaxations of the mixed integer programs. Section 4 presents three formulations of DSP and in Section 5 these formulations are compared. In Section 6 computational experiments are reported about the solution of DSP through these formulations. Section 7 is a conclusion.

2. Mixed integer linear programming formulations for HSP

In this section we present 4 formulations of HSP. It is not difficult to check that these formulations are valid for DSP by putting $w_{i,j} = 1$ for all $[v_i, v_j] \in E$. First we define MIP1, the standard mixed integer linear reformulation of Q. We substitute the variables $y_{i,j}$ for the products $x_i x_j$ and we add $2|E|$ linearization constraints.

$$
\text{(MIP1)} \quad \begin{cases} \text{Maximize } f_{L_1}(y) = \sum_{(i,j)\in T} w_{i,j} y_{ij} & \\ \text{s.t.} & \\ \sum_{i=1}^{n} x_i = k & (1) \\ y_{i,j} \le x_i \qquad (i,j) \in T & (3) \\ y_{i,j} \le x_j \qquad (i,j) \in T & (4) \\ y_{i,j} \ge 0 \qquad (i,j) \in T & (5) \\ x_i \in \{0,1\} \qquad (i = 1,...,n) & (2) \end{cases}
$$

MIP1 contains n 0-1 variables, $|E|$ positive variables and $2|E|+1$ constraints without counting the non-negativity ones. This linearization of quadratic 0-1 programs was initially proposed by Rhys [Rhys, 1970] and was extensively studied in the unconstrained case by Hammer, Hansen and Simeone [Hammer et al., 1984].

In the second considered linearization of HSP, MIP2, there are $O(n)$ variables and $O(n)$ constraints.

$$
\text{(MIP2)} \quad
\begin{cases}
\text{Maximize } f_{L_2}(z) = \dfrac{1}{2}\sum_{i=1}^{n} z_i & \\
\text{s.t.} & \\
\sum_{i=1}^{n} x_i = k & (1) \\
z_i \le \sum_{j:(j,i)\in T} w_{j,i} x_j + \sum_{j:(i,j)\in T} w_{i,j} x_j \qquad (i = 1,...,n) & (6) \\
z_i \le (\sum_{j:(j,i)\in T} w_{j,i} + \sum_{j:(i,j)\in T} w_{i,j}) x_i \qquad (i = 1,...,n) & (7) \\
z_i \ge 0 \qquad (i = 1,...,n) & (8) \\
x_i \in \{0,1\} \qquad (i = 1,...,n) & (2)
\end{cases}
$$

MIP2 may be considered as a variant of the linearization proposed in [Glover, 1975] for the general problem of optimizing a 0-1 quadratic function subject to linear constraints, adapted to the particular case of HSP. In this linearization, there are n binary variables, n positive variables and $2n+1$ constraints. In fact we will consider a slightly different formulation of HSP which takes account of obvious upper and lower bounds for the quantity $\sum_{j:(j,i)\in T} w_{j,i} x_j + \sum_{j:(i,j)\in T} w_{i,j} x_j$, $(i = 1,...,n)$, under the constraint $\sum_{i=1}^{n} x_i = k$. It is easy to check that if $x_i = 1$, $\sum_{j:(j,i)\in T} w_{j,i} x_j + \sum_{j:(i,j)\in T} w_{i,j} x_j$ is greater than or equal to the sum of the $k-1$ smallest values of the set $W_i = \{w_{i,j} : (i,j) \in T\} \cup \{w_{j,i} : (j,i) \in T\}$, and less than or equal to the sum of the $k-1$ greatest values of the same set. Denoting respectively by $L(W_i)$ and $U(W_i)$ these two sums we obtain the mixed integer linear program MIP2'.

$$
\text{(MIP2')} \quad
\begin{cases}
\text{Maximize } f'_{L_2}(x) = \dfrac{1}{2}\sum_{i=1}^{n} L(W_i) x_i + \dfrac{1}{2}\sum_{i=1}^{n} t_i & \\
\text{s.t.} & \\
\sum_{i=1}^{n} x_i = k & (1) \\
t_i \le \sum_{j:(j,i)\in T} w_{j,i} x_j + \sum_{j:(i,j)\in T} w_{i,j} x_j - L(W_i) \qquad (i = 1,...,n) & (9) \\
t_i \le (U(W_i) - L(W_i)) x_i \qquad (i = 1,...,n) & (10) \\
t_i \ge 0 \qquad (i = 1,...,n) & (11) \\
x_i \in \{0,1\} \qquad (i = 1,...,n) & (12)
\end{cases}
$$

Computational experiments have shown that formulation MIP2' was much more efficient than formulation MIP2.

Proposition 1 Problems Q and MIP2' are equivalent in the following sense: given any feasible solution $\tilde{x} = (\tilde{x}_1,...,\tilde{x}_n)$ in Q, there exists a vector $\tilde{t} = (\tilde{t}_1,...,\tilde{t}_n)$ such that $(\tilde{x},\tilde{t})$ is feasible in MIP2', with the same objective value. Conversely, given any optimal solution $(\tilde{x},\tilde{t})$ in MIP2', the corresponding solution $\tilde{x}$ is feasible in Q, with the same objective value.

Proof

Let $\tilde{x} = (\tilde{x}_1,...,\tilde{x}_n)$ be a feasible solution of Q. Its value is equal to $\sum_{(i,j)\in T} w_{i,j}\tilde{x}_i\tilde{x}_j$. Consider the solution $(\tilde{x},\tilde{t})$ where $\tilde{t}_i = \tilde{x}_i(\sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j - L(W_i))$ for all $i \in \{1,...,n\}$. Obviously constraints (9) and (11) are satisfied. On the other hand, since when $x_i = 1$, $\sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j \le U(W_i)$, we get $\tilde{t}_i \le \tilde{x}_i(U(W_i) - L(W_i))$ and constraint (10) is satisfied. $(\tilde{x},\tilde{t})$ is therefore a feasible solution to (MIP2'). Its value is equal to $\frac{1}{2}\sum_{i=1}^{n} L(W_i)\tilde{x}_i + \frac{1}{2}\sum_{i=1}^{n}(\sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j - L(W_i))\tilde{x}_i =$ $\sum_{(i,j)\in T} w_{i,j}\tilde{x}_i\tilde{x}_j$. Conversely, let $(\tilde{x},\tilde{t})$ be an optimal solution to MIP2'. Obviously $\tilde{x}$ is a feasible solution of Q and if $\tilde{x}_i = 0$ then $\tilde{t}_i = 0$ because of constraints (10)-(11). Since the objective function has to be maximized, if $\tilde{x}_i = 1$ then the variable t_i takes the greatest possible value, i.e. $\tilde{t}_i = \sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j - L(W_i)$. Finally we get $\tilde{t}_i = \tilde{x}_i(\sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j - L(W_i))$ and the objective value in MIP2' corresponding to the optimal solution $(\tilde{x},\tilde{t})$ is equal to $\frac{1}{2}\sum_{i=1}^{n} L(W_i)\tilde{x}_i + \frac{1}{2}\sum_{i=1}^{n}(\sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j - L(W_i))\tilde{x}_i =$ $\sum_{(i,j)\in T} w_{i,j}\tilde{x}_i\tilde{x}_j$, the objective value associated with solution $\tilde{x}$ of Q.

The linearization technique allowing to formulate HSP as MIP3 is well known. It was proposed in [Adams and Sherali, 1986] for the general linearly constrained 0-1 quadratic programming problem and it is proved in [Billionnet and Faye, 1997] that the optimal value of the continuous relaxation of MIP3 is equal to the value of the greatest

constant c such that there exist a quadratic posiform ϕ satisfying $f = c + \phi$ for all $x \in \{0,1\}^n$ with $\sum_{i=1}^{n} x_i = k$ ($\phi = c_1 T_1 + c_2 T_2 + ... + c_m T_m$, where each term T_i is a literal ,i.e. x_i or $\bar{x}_i = 1 - x_i$, or a product of two literals and the c_i are all positive, is called a quadratic posiform). Earlier, this linearization technique was used in [Frieze and Yadegar, 1983] for the quadratic assignment problem. This formulation contains n 0-1 variables, $n(n-1)/2$ positive variables, and $2|E| + n + 1$ constraints. As in MIP1, the variables $y_{i,j}$ correspond to the products $x_i x_j$.

$$
\text{(MIP3)} \quad
\begin{cases}
\text{Maximize } f_{L_3}(x) = \sum_{(i,j)\in T} w_{i,j} y_{i,j} & & \\
\text{s.t.} & & \\
\sum_{i=1}^{n} x_i = k & & (1) \\
\sum_{j<i} y_{j,i} + \sum_{j>i} y_{i,j} = (k-1) x_i & (i = 1,...,n) & (12) \\
y_{i,j} \le x_i & ((i,j) \in T) & (13) \\
y_{i,j} \le x_j & ((i,j) \in T) & (14) \\
y_{i,j} \ge 0 & ((i,j) : i < j) & (15) \\
x_i \in \{0,1\} & (i = 1,...,n) & (2)
\end{cases}
$$

MIP3 is obtained by adding constraints (12) to MIP1. These constraints are obtained by multiplying both sides of the constraint $\sum_{i=1}^{n} x_i = k$ by x_i, $i = 1,...,n$, and then by linearizing the obtained equalities. Note that in this formulation all the products $x_i x_j, i < j$, and therefore all the variables $y_{i,j}$ have to be considered

3. Theoretical comparison of the formulations of HSP

It is well-known that the effectiveness of the resolution of a combinatorial optimization problem by (mixed) integer linear programming strongly depends on the quality of the continuous relaxation ($0 \le x_i \le 1$ in place of $x_i \in \{0,1\}$) of the considered program. In this section we compare continuous relaxations of the programs MIP1, MIP2, MIP2' and MIP3 which we will denote by $\overline{\text{MIP1}}, \overline{\text{MIP2}}, \overline{\text{MIP2'}}$, and $\overline{\text{MIP3}}$, respectively. For a mathematical program Π we will denote by opt(Π) its optimal value.

Proposition 2

(i) $\text{opt}(\overline{\text{MIP2'}}) \leq \text{opt}(\overline{\text{MIP2}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) Let $(\tilde{x}, \tilde{t})$ be a feasible solution of $\overline{\text{MIP2'}}$. Its value is equal to $\frac{1}{2}(\sum_{i=1}^{n} L(W_i)\tilde{x}_i + \sum_{i=1}^{n} \tilde{t}_i)$. From this solution let us build a solution to $\overline{\text{MIP2}}$: $(\tilde{x}, \tilde{z})$ such that $\tilde{z}_i = \tilde{t}_i + L(W_i)\tilde{x}_i$ for all i $\in \{1,...,n\}$. Obviously, the value of this solution is also equal to $\frac{1}{2}(\sum_{i=1}^{n} L(W_i)\tilde{x}_i + \sum_{i=1}^{n} \tilde{t}_i)$. So, we have just to prove that $(\tilde{x}, \tilde{z})$ is a feasible solution to $\overline{\text{MIP2}}$. Constraint (9) in $\overline{\text{MIP2'}}$ implies $\tilde{t}_i + L(W_i) \leq \sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j$ $+ \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j \Rightarrow \tilde{t}_i + L(W_i)\tilde{x}_i \leq \sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j$ since $0 \leq x_i \leq 1$ and $L(W_i) \geq 0$. Constraint (6) of $\overline{\text{MIP2}}$ is satisfied. Constraints (10) in $\overline{\text{MIP2'}}$ imply $\tilde{t}_i + L(W_i)\tilde{x}_i \leq U(W_i)\tilde{x}_i$ and since, by definition, $U(W_i) \leq \sum_{j:(j,i)\in T} w_{j,i} + \sum_{j:(i,j)\in T} w_{i,j}$, we get $U(W_i)\tilde{x}_i \leq (\sum_{j:(j,i)\in T} w_{j,i} + \sum_{j:(i,j)\in T} w_{i,j})\tilde{x}_i$ and constraint (7) of $\overline{\text{MIP2}}$ is satisfied.

(ii) Consider the instance corresponding to a complete graph of 4 vertices with *k*=3. The values associated with the edges are: $w_{1,2} = 5, w_{1,3} = 9, w_{1,4} = 9, w_{2,3} = 0, w_{2,4} = 6$ and $w_{3,4} = 6$. We obtain for this instance $\text{opt}(\overline{\text{MIP2}}) = 26.25$ with $x = (0.75, 0.75, 0.75, 0.75)$ and $\text{opt}(\overline{\text{MIP2'}}) = 24$ with $x = (1,0,1,1)$.

Proposition 3

(i) $\text{opt}(\overline{\text{MIP1}}) \leq \text{opt}(\overline{\text{MIP2}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) Let $(\tilde{x}, \tilde{y})$ be a feasible solution of $(\overline{\text{MIP1}})$. Its value is equal to $\sum_{(i,j)\in T} w_{i,j}\tilde{y}_{i,j}$. From this solution let us build a solution to $(\overline{\text{MIP2}})$: $(\tilde{x}, \tilde{z})$ such that $\tilde{z}_i = \sum_{j:(j,i)\in T} w_{j,i}\tilde{y}_{i,j} + \sum_{j:(i,j)\in T} w_{i,j}\tilde{y}_{i,j}$ for all $i \in \{1,...,n\}$. The value of this solution is equal to $\frac{1}{2}\sum_{i=1}^{n} \tilde{z}_i = \sum_{(i,j)\in T} w_{i,j}\tilde{y}_{i,j}$. So we have just to prove that $(\tilde{x}, \tilde{z})$ is a feasible solution to $(\overline{\text{MIP2}})$. Since $\tilde{y}_{i,j} \le \tilde{x}_j$ for all $(i,j)\in T$ we get $\tilde{z}_i = \sum_{j:(j,i)\in T} w_{j,i}\tilde{y}_{i,j} + \sum_{j:(i,j)\in T} w_{i,j}\tilde{y}_{i,j} \le \sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_j + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_j$ and constraint (6) of $\overline{\text{MIP2}}$ is satisfied. Since $\tilde{y}_{i,j} \le \tilde{x}_i$ for all $(i,j)\in T$, we get $\tilde{z}_i = \sum_{j:(j,i)\in T} w_{j,i}\tilde{y}_{i,j} + \sum_{j:(i,j)\in T} w_{i,j}\tilde{y}_{i,j} \le \sum_{j:(j,i)\in T} w_{j,i}\tilde{x}_i + \sum_{j:(i,j)\in T} w_{i,j}\tilde{x}_i$ and constraint (7) of $\overline{\text{MIP2}}$ is satisfied.

(ii) Consider the graph of 6 vertices with the edges $[v_1, v_2]$, $[v_1, v_3], [v_1, v_4], [v_2, v_4], [v_4, v_6], [v_5, v_6]$ and k=3. All these edges have a value equal to 1. We obtain for this instance $\text{opt}(\overline{\text{MIP2}}) = 3.25$ with $x = (0.625, 0.875, 0, 1, 0, 0.5)$ and $\text{opt}(\overline{\text{MIP1}}) = 3$ with $x = (1,1,0,1,0,0)$.

Proposition 4 $\overline{\text{MIP2'}}$ and $\overline{\text{MIP3}}$ are not comparable in the sense that some HSP instances are such that $\overline{\text{MIP2'}}$ yields a tighter upper bound (of lower value) that the one given by $\overline{\text{MIP3}}$, while other instances can yield the inverse result.

Proof

This result can be checked in Tables 1.a and 1.b. Indeed for a density of 25% and k=10 the relative gap associated with $\overline{\text{MIP2'}}$ is equal to 59.4% while the one associated with $\overline{\text{MIP3}}$ is equal to 55%. Conversely, when the density is 50% and k=30 the relative gap associated with $\overline{\text{MIP2'}}$ is equal to 4.7% while the one associated with $\overline{\text{MIP3}}$ is equal to 15.2%.

Proposition 5 $\overline{\text{MIP1}}$ and $\overline{\text{MIP2'}}$ are not comparable in the sense that some HSP instances are such that $\overline{\text{MIP1}}$ yields a tighter upper bound (of lower value) that the one given by $\overline{\text{MIP2'}}$, while other instances can yield the inverse result.

Proof

One can check in Table 1.a that, for a density of 25% and k=10, the relative gap associated with $\overline{\text{MIP1}}$ is equal to 76.3% while the one associated with $\overline{\text{MIP2'}}$ is equal to 59.4%. Conversely, consider a graph of 6 nodes with the edges $[v_1, v_4]$, $[v_1, v_5]$, $[v_2, v_3]$, $[v_2, v_5]$, $[v_4, v_6]$, and $[v_5, v_6]$ of value 1, and k=4. For this particular instance $\text{opt}(\overline{\text{MIP1}}) = 4$ with $x \approx (0.67, 0.67, 0.67, 0.67, 0.67, 0.67)$ and $\text{opt}(\overline{\text{MIP2'}}) = 4.17$ with $x \approx (0.89, 0.44, 0, 0.89, 0.89, 0.89, 0.89)$.

Proposition 6

(i) $\text{opt}(\overline{\text{MIP3}}) \leq \text{opt}(\overline{\text{MIP1}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) is obvious since MIP3 is obtained by adding supplementary constraints to MIP1. One can check in Tables 1.a and 1.b that (ii) is true since, for a density of 25% and a value of k equal to 10, the relative gaps for MIP1 and MIP3 are 76.3% and 55%, respectively.

Figure 1 summarizes the relationships between the four relaxations $\overline{\text{MIP1}}$, $\overline{\text{MIP2}}$, $\overline{\text{MIP2'}}$ and $\overline{\text{MIP3}}$.

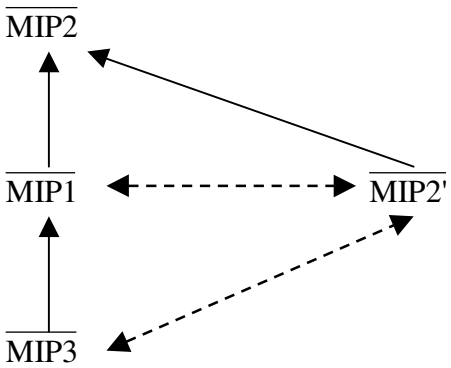

Figure 1. Relationships between different relaxations of HSP
a ⟶ b : formulation "a" is tighter than formulation "b"
a ⟷ b : formulations "a" and "b" are not comparable

4. Mixed integer linear programming formulations of DSP

In this section we present 3 formulations which are specific of DSP. They do not allow to model HSP directly. MIP4 consists in minimizing the number of edges that are not in the *k*-subgraph. Let $\overline{T} = \left\{(i, j) \in \{1,...,n\}^2 : i < j, [v_i, v_j] \notin E\right\}$.

$$
(\text{MIP4}) \begin{cases} \text{Maximize} \quad f_{L_4}(y) = \dfrac{1}{2} k(k-1) - \sum_{(i,j)\in\overline{T}} y_{i,j} & \\ \text{s.t.} & \\ \sum_{i=1}^{n} x_i = k & (1) \\ x_i + x_j \le 1 + y_{i,j} \quad (i,j) \in \overline{T} & (16) \\ y_{i,j} \ge 0 \quad (i,j) \in \overline{T} & (17) \\ x_i \in \{0,1\} \quad (i = 1,...,n) & (2) \end{cases}
$$

The constraints $x_i + x_j \le 1 + y_{i,j}$ express the fact that if both vertices v_i and v_j are in the *k*-subgraph, then the variable $y_{i,j}$ must take 1 as value which means that the edge $[v_i, v_j]$ misses compared to a complete *k*-subgraph. The experiments showed that this formulation is particularly interesting in the case of dense graphs. Indeed the number of variables in MIP4 is equal to $n + n(n-1)/2 - |E|$ and the number of constraints is $1 + n(n-1)/2 - |E|$.

Formulation MIP5 consists, as MIP4, in minimizing the number of edges that are not in the *k*-subgraph but the number of variables in MIP5 is $2n$ and the number of constraints is $1+n$. For all i, d_i is the degree of vertex v_i and constraints (18) express the fact that if vertex v_i is in the *k*-subgraph S, i.e. $x_i = 1$, then the set of edges $E_i = \left\{[v_i, v_j] : v_j \in S, [v_i, v_j] \text{ or } [v_j, v_i] \in \overline{T}\right\}$ misses compared to a complete *k*-subgraph. Due to the fact that one minimizes the quantity $-\frac{1}{2}\sum_{i=1}^{n} u_i$, u_i will be equal to the smallest possible value, i.e. the cardinality of E_i, in all optimal solutions of MIP5.

$$
(\text{MIP5})\begin{cases}
\text{maximize } f_{L_5}(y) = \dfrac{1}{2}k(k-1) - \dfrac{1}{2}\sum_{i=1}^{n} u_i & & \\
\text{s.t.} & & \\
\sum_{i=1}^{n} x_i = k & & (1)\\
(n-1-d_i)x_i + \sum_{j:(i,j) \text{ or } (j,i)\in \overline{T}} x_j \le n-1-d_i+u_i & (i=1,...,n) & (18)\\
u_i \ge 0 & (i=1,...,n) & (19)\\
x_i \in \{0,1\} & (i=1,...,n) & (2)
\end{cases}
$$

Formulation MIP6 consists in adding to MIP4 $O(n^3)$ cuts : $x_i + x_j + x_k \le 1 + y_{i,j} + y_{j,k} + y_{i,k}$ for all (i,j,k) such that $i<j<k$ and the three edges $[v_i, v_j], [v_j, v_k], [v_i, v_k]$ are not present in the graph G. One can easily check that these constraints are valid inequalities. They express the fact that if two vertices among x_i, x_j, x_k are in the *k*-subgraph then $y_{i,j} + y_{j,k} + y_{i,k} \ge 1$ and if the three vertices x_i, x_j, x_k are in the *k*-subgraph then $y_{i,j} + y_{j,k} + y_{i,k} \ge 2$.

$$
(\text{MIP6})\begin{cases}
\text{Maximize} \quad f_{L_6}(y) = \dfrac{1}{2}k(k-1) - \sum_{(i,j)\in \overline{T}} y_{i,j} & & \\
\text{s.t.} & & \\
\sum_{i=1}^{n} x_i = k & & (1)\\
x_i + x_j \le 1 + y_{i,j} & (i,j)\in \overline{T} & (16)\\
y_{i,j} \ge 0 & (i,j)\in \overline{T} & (17)\\
x_i + x_j + x_k \le 1 + y_{i,j} + y_{j,k} + y_{i,k} & ((i,j),(j,k),(i,k)\in \overline{T}) & (20)\\
x_i \in \{0,1\} & (i=1,...,n) & (2)
\end{cases}
$$

5. Theoretical comparison of the formulations of DSP

We compare in this section continuous relaxations of programs MIP4, MIP5, and MIP6 which we will denote by $\overline{\text{MIP4}}, \overline{\text{MIP5}},$ and $\overline{\text{MIP6}}$, respectively. Recall that the optimal value of the program Π is denoted by opt(Π).

Proposition 7

(i) $\text{opt}(\overline{\text{MIP4}}) \le \text{opt}(\overline{\text{MIP5}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) Let $(\tilde{x}, \tilde{y})$ be a feasible solution of $\overline{\text{MIP4}}$. Its value is equal to $\frac{1}{2}k(k-1) - \sum_{(i,j)\in\overline{T}} \tilde{y}_{i,j}$. From this solution let us build the following solution to $\overline{\text{MIP5}}$: $(\tilde{x}, \tilde{u})$ such that $\tilde{u}_i = \sum_{j:(i,j)\in\overline{T}} \tilde{y}_{i,j} + \sum_{j:(i,j)\in\overline{T}} \tilde{y}_{j,i}$ for all $i \in \{1,...,n\}$. Obviously, the value of this solution is also equal to $\frac{1}{2}k(k-1) - \sum_{(i,j)\in\overline{T}} \tilde{y}_{i,j}$. So, we have just to prove that $(\tilde{x}, \tilde{u})$ is a feasible solution to $\overline{\text{MIP5}}$, i.e. that constraint (18) is satisfied. Let $\overline{d}_i = n-1-d_i$.

(18) $\Leftrightarrow (n-1-d_i)x_i + \sum_{j:(i,j) \text{ or } (i,j)\in\overline{T}} x_j \leq n-1-d_i + \sum_{j:(i,j)\in\overline{T}} \tilde{y}_{i,j} + \sum_{j:(j,i)\in\overline{T}} \tilde{y}_{j,i}$

$\Leftrightarrow (n-1-d_i)x_i + \sum_{j:(i,j) \text{ or } (i,j)\in\overline{T}} x_j \leq n-1-d_i + \sum_{j:(i,j) \text{ or } (j,i)\in\overline{T}} x_j + \overline{d}_i x_i - \overline{d}_i$

$\Leftrightarrow (n-1)x_i - (d_i + \overline{d}_i)x_i \leq (n-1) - (d_i + \overline{d}_i)$. This last inequality is true since $(d_i + \overline{d}_i) = n-1$.

(ii) One can check in Tables 1.b and 1.c that for a density of 50% and *k*=30 the relative gap associated with $\overline{\text{MIP4}}$ is equal to 4.9% while the one associated with $\overline{\text{MIP5}}$ is equal to 8.4%.

Proposition 8

(i) $\text{opt}(\overline{\text{MIP6}}) \leq \text{opt}(\overline{\text{MIP5}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) It is a direct consequence of Proposition 7 since (MIP6) is built by adding constraints to (MIP4)

(ii) It is easy to verify (ii) by observing in Table 1.c that for a density of 50% and *k*=30 the relative gap of $\overline{\text{MIP5}}$ is equal to 8.4% while the relative gap of $\overline{\text{MIP6}}$ is equal to 0.5%.

Proposition 9

(i) $\text{opt}(\overline{\text{MIP6}}) \leq \text{opt}(\overline{\text{MIP4}})$;

(ii) there exists some instances for which this inequality is strict.

Proof

(i) it is obvious since MIP6 is built by adding cuts to MIP4;

(ii) We can see in Tables 1.b and 1.c that, for a density of 75% and a value of k equal to 20, we get a relative gap of 11.3% for $\overline{\text{MIP4}}$ and a relative gap of 0.25% for $\overline{\text{MIP6}}$.

Figure 2 summarizes the relationships between the linearizations of DSP.

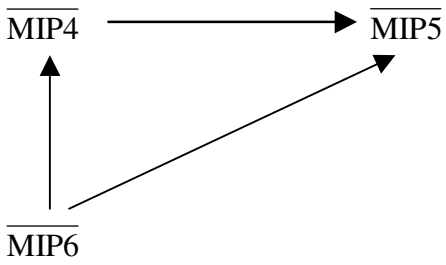

Figure 2. Relationships between different relaxations of DSP
a ⟶ b : formulation "a" is tighter than formulation "b"

6. Computational results

The experiments have been carried out on the dense subgraph problem (all the edges have the same weight). The six programs MIP1, MIP2', MIP3, MIP4, MIP5 and MIP6 have been solved using XA solver [XA, 1994] on a pentium II 300Mhz computer. The experiments have been performed on randomly generated graphs. In Table 1 we compare the six mixed integer linearizations on instances with 40 nodes (n=40) for three values of the graph density (d=25%, 50%, 75%), and three values of k (n/4, n/2, 3n/4) (the density of a graph with n vertices and $|E|$ edges is equal to $2|E|/n(n-1)$). Each line of this table gives average results for 5 randomly generated instances. When the 5 instances could not be exactly solved within the time limit, fixed to 900 s. (15 minutes), the presented results only concern the solved instances.

		MIP 1				MIP 2'			
density	k	Relative gap (%)	# nodes	CPU time (s)	# solved instances	Relative gap (%)	# nodes	CPU time (s)	# solved instances
0.25	10	76.3	5046.8	119.8	5	59.4	109977	411.7	4
	20	27.2	3024.8	76.6	5	26.9	74775	334.7	4
	30	7.3	223.6	7.0	5	7.1	2622.8	11.0	5
0.50	10					12.5	32377	105.0	5
	20								
	30	16.0	3120.2	242.2	4	4.7	1568.2	6.0	5
0.75	10					0.0	92.0	1.0	5
	20								
	30					3.2	1289.2	4.8	5

Table 1.a

		MIP 3				MIP 4			
density	k	Relative gap (%)	# nodes	CPU time (s)	# solved instances	Relative gap (%)	# nodes	CPU time (s)	# solved instances
0.25	10	55.0	3180.8	326.4	5	22.9	17665	525.6	5
	20	25.1	1262.0	262.2	4				
	30	7.3	260.6	44.0	5	18.6	1752.0	68.4	5
0.50	10	12.5	840.7	515.0	3	12.5	984.2	18.4	5
	20					46.2	14322	556.0	1
	30	15.2	1787.0	647.0	1	4.9	74.4	2.6	5
0.75	10	0.0	11.2	8.8	5	0.0	20.6	1.0	5
	20					11.3	390.2	7.2	5
	30					1.2	14.2	1.0	5

Table 1.b

		MIP 5				MIP 6			
density	k	Relative gap (%)	# nodes	CPU time (s)	# solved instances	Relative gap (%)	# nodes	CPU time (s)	# solved instances
0.25	10								
	20								
	30	23.9	85733	161.4	5				
0.50	10	12.5	24088	47.0	5	12,5	166.0	52,8	5
	20					3.9	34.0	127.6	5
	30	8.4	2566.6	4.8	5	0.5	4.4	26.2	5
0.75	10	0.0	271.4	0.3	5	0.0	3.8	0.1	5
	20	10.7	140577	429.7	3	0.25	5.0	0.7	5
	30	3.2	343.6	0.6	5	0.0	0.2	0.4	5

Table 1.c

Table 1.a, 1.b and 1.c. Numerical comparison between MIP1, MIP2', MIP3, MIP4, MIP5 and MIP6 for the dense *k*-subgraph problem on randomly generated graphs with 40 vertices. For a mathematical program Π, the relative gap is equal to the ratio $(opt(\overline{\Pi}) - opt(\Pi))/opt(\Pi)$ where $\overline{\Pi}$ is the continuous relaxation of Π; # nodes is the number of nodes considered in the search tree of the branch and bound procedure.

: 5 instances out of 5 are solved in less than 900s
: 4 instances out of 5 are solved in less than 900s
: 3 instances out of 5 are solved in less than 900s
: 2 instances out of 5 are solved in less than 900s
: 1 instance out of 5 is solved in less than 900s
: none of the 5 instances is solved in less than 900s

For *d*=0.25, only MIP1 allows all instances to be solved in less than 15 minutes of CPU time. However, for *k*=30, MIP2', MIP3, MIP4 and MIP5 are also efficient formulations.

For *d*=0.50, only MIP6 allows all the instances to be solved. None of the other methods allows the considered instances with *k*=20 to be solved in less than 15 minutes of CPU time (only one instance over 5 is solved by MIP4).

For *d*=0.75, MIP4 and MIP6 allow all the considered instances to be solved. For this density, MIP1 is a bad choice since none of the 15 instances can be solved in this way. However, for *k*=10, MIP2', MIP3 and MIP5 are also efficient formulations and, for *k*=30, the other interesting formulations are MIP2' and MIP5. Table 2 summarizes the numerical comparisons between all the linearizations, with regard to CPU time, by presenting the

formulation recommended for solving DSP taking into account d, the density of the graph and k, the number of nodes in the subgraph.

	d=0.25	d=0.50	d=0.75
k=10	MIP1 (120 s.)	MIP4 (18 s.)	MIP2' (1 s.) MIP4 (1 s.) MIP5 (0.3) MIP6 (0.1)
k=20	MIP1 (77 s.)	MIP6 (128 s.)	MIP6 (0.7 s.)
k=30	MIP1 (7 s.) MIP2' (11 s.)	MIP2' (6 s.) MIP4 (3 s.) MIP5 (5 s.)	MIP2' (5 s.) MIP4 (1 s.) MIP5 (0.6) MIP6 (0.4)

Table 2. Mixed integer linear formulations recommended for solving DSP (n=40) and associated CPU times

7. Conclusion

There are in general several formulations of the same combinatorial optimization problem by mixed integer linear programming. In this paper we proposed 3 different formulations of DSP and 4 different formulations of HSP which also are formulations of DSP. The experiments showed that no formulation can be regarded as the best one and that the effectiveness of a formulation strongly depends on the instance structure. As shown in Table 2, according to the type of considered instance (density of the graph and number of vertices in the subgraph), it is necessary to choose one or the other formulation. For example, MIP1 which is the best formulation for a graph of density 25% proves to be very bad for the densities 50% and 75%. Table 2 shows that all the considered instances with 40 vertices (density equal to 25%, 50%, 75% and k equal to 10, 20, 30) can be solved provided that one chooses the good formulation. The most difficult instances of DSP seem to be, at least for this approach by mixed integer linear programming, the graphs of density 50% with k=n/2. As it is well known, the solution of a combinatorial optimization problem by this approach has many advantages compared to specific algorithms: simplicity of implementation, robustness of MIP professional software and possibility of easily adding new constraints to the problem. On the other hand, the results obtained in this work confirm that an important difficulty that arises when solving a combinatorial optimization problem by using mixed linear programming is the choice of a good formulation. Indeed it

seems that this choice is often difficult to make before a large number of experiments have been carried out

References

W.P.Adams and H.D.Sherali, A tight linearization and an algorithm for zero-one quadratic programming problems. Management Science, vol.32, n°10, October 1986, 1274-1290.

Y.Asahiro, K.Iwama, H.Tamaki and T.Tokuyama, Greedily finding a dense subgraph. Proceedings of the 5th Scandinavian Workshop on Algorithm Theory. Lectures notes in Computer Science, 1097, pp. 136-148, Springer-Verlag, 1996.

E.M.L.Beale, Introduction to optimization, Willey, New-York, 1988, 121p.

A.Billionnet and A.Faye, A lower bound for a constrained quadratic 0-1 minimization problem. Discrete Applied Mathematics, 74, 1997, pp. 135-146.

A.Billionnet, A.Faye and E.Soutif, A new upper bound for the 0-1 quadratic knapsack problem. European Journal of Operational Research, 112/3, 1999, pp. 664-672.

A.Caprara, D.Pisinger and P.Toth, Exact solution of the quadratic knapsack problem, INFORMS Journal on Computing, 11, 1999, pp.125-137.

D.G.Corneil and Y.Perl, Clustering and domination in perfect graphs. Discrete Applied Mathematics, vol.9, No. 1, 1984, 27-39.

E.Erkut, The discrete p-dispersion problem. European Journal of Operational Research, 46, 1990, pp. 46-80.

A.M.Frieze and J.Yadegar, On the quadratic assignment problem. Discrete Applied mathematics, 5, 1983, pp.89-98.

F.Glover, Improved linear integer programming formulations of nonlinear integer problems, Management Science, vol.22, 1975, 455-460.

P.L.Hammer, P.Hansen and B.Simeone, Roof duality, complementation and persistency in quadratic 0-1 optimization. Math. Programming, 28 (1984), pp. 121-155.

P.L.Hammer and D.J.Rader, Efficient methods for solving 0-1 quadratic knapsack problems. INFOR 35, No. 3, 1997, pp.170-182.

R. Hassin, S.Rubinstein and A.Tamir, Approximation algorithms for maximum dispersion. Technical Report, Department of Statistics and Operations Research, Tel Aviv University, June 1997.

R.K.Kincaid, Good solutions to discrete noxious location problems via metaheuristics. Annals of Operations Research, 40, 1992, pp. 265-281.

G.Kortsarz and D.Peleg, On choosing a dense subgraph. Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, pp. 692-701, 1993.

J.Rhys, A selection problem of shared fixed costs and networks. Management science17, 1970, 200-207.

H.M.Salkin and K.Mathur, Foundation of integer programming, North-Holland, Amsterdam, 1989, 755p.

A.Srivastav and K.Wolf, Finding dense subgraph with semidefiinite programming. Research Report No. 97.301, Angewandte Mathematik und Informatik Universitt zu Kuln, 1997.

[XA, 1994] XA Callable Library, Sunset Software Technology, 1994, San Marino, CA, USA.