On the Hitting Set of Bundles Problem

E. Angel, E. Bampis, Laurent Gourvès

UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS UMR 7024

CAHIER DU LAMSADE
259

Juillet 2007

On the Hitting Set of Bundles Problem

E. Angel, E. Bampis, L. Gourvès

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

On the Hitting Set of Bundles Problem

Eric Angel*, Evripidis Bampis*, Laurent Gourvès‡

Résumé

Le problème de l'ensemble minimal de paquets (*minimal hitting set of bundles problem* ou HSB) est défini comme suit. On dispose d'un ensemble $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ de n éléments. Chaque élément e_i ($i = 1, \ldots, n$) a un coût positif ou nul c_i. Un *paquet* b est un sous ensemble de $\mathcal{E}$. On dispose aussi d'une collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ de m ensembles de paquets. De manière plus précise, chaque ensemble S_j ($j = 1, \ldots, m$) est composé de $g(j)$ paquets distincts notés $b_j^1, b_j^2, \ldots, b_j^{g(j)}$. Une solution du problème HSB est un sous ensemble $\mathcal{E}' \subseteq \mathcal{E}$ tel que pour tout $S_j \in \mathcal{S}$, au moins un paquet est couvert, *i.e.* $b_j^l \subseteq \mathcal{E}'$. Le *coût total* de la solution, noté $C(\mathcal{E}')$, est $\sum_{\{i|e_i \in \mathcal{E}'\}} c_i$. Le problème consiste à trouver une solution de coût total minimum. Nous donnons un algorithme déterministe $N(1 - (1 - \frac{1}{N})^M)$-approché, où N est le nombre maximal de paquets par ensemble et M est le nombre maximal d'ensembles à qui un élément appartient. Le rapport d'approximation est à peu de choses près le meilleur que l'on puisse proposer car on peut montrer que HSB ne peut être approché avec un rapport $7/6 - \epsilon$ lorsque $N = 2$ et $N - 1 - \epsilon$ lorsque $N \geq 3$. L'algorithme proposé est aussi le premier offrant une garantie de performance pour le problème classique d'optimisation de requêtes multiples [9, 10]. Son rapport d'approximation pour le problème MIN $k-$SAT dont il est une généralisation est le même que celui du meilleur algorithme connu [3].

Mots-clefs : optimisation combinatoire, algorithme d'approximation, problème HITTING SET

Résumé

The minimum HITTING SET OF BUNDLES problem (HSB) is defined as follows. We are given a set $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ of n elements. Each element e_i

*IBISC, Université d'Evry Val d'Essonne, 91000 Evry, France. {angel, bampis}@ibisc.fr

‡LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. laurent.gourves@lamsade.dauphine.fr

$(i = 1, \ldots, n)$ has a non negative cost c_i. A *bundle* b is a subset of $\mathcal{E}$. We are also given a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of m sets of bundles. More precisely, each set S_j $(j = 1, \ldots, m)$ is composed of $g(j)$ distinct bundles $b_j^1, b_j^2, \ldots, b_j^{g(j)}$. A solution to HSB is a subset $\mathcal{E}' \subseteq \mathcal{E}$ such that for every $S_j \in \mathcal{S}$ at least one bundle is covered, i.e. $b_j^l \subseteq \mathcal{E}'$. The *total cost* of the solution, denoted as $C(\mathcal{E}')$, is $\sum_{\{i|e_i \in \mathcal{E}'\}} c_i$. The problem is to find a solution with *minimum* total cost.

We give a deterministic $N(1-(1-\frac{1}{N})^M)$-approximation algorithm, where N is the maximal number of bundles per set and M is the maximal number of sets an element can appear in. This is roughly speaking the best approximation ratio that we can obtain for the HSB problem since we also prove that HSB cannot be approximated within $7/6-\epsilon$ when $N=2$ and $N-1-\epsilon$ when $N \geq 3$. Our algorithm is also the first approximation algorithm with guaranteed performance for the classical MULTIPLE-QUERY OPTIMIZATION problem [9, 10], while it matches the best approximation ratio for the MIN $k-$SAT problem (for general k) obtained by the algorithm of [3].

Key words : combinatorial optimization, approximation algorithm, HITTING SET problem

1 Introduction

The minimum HITTING SET OF BUNDLES problem (HSB) is defined as follows. We are given a set $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ of n elements. Each element e_i $(i = 1, \ldots, n)$ has a non negative cost c_i. A *bundle* b is a subset of $\mathcal{E}$. We are also given a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of m sets of bundles. More precisely, each set S_j $(j = 1, \ldots, m)$ is composed of $g(j)$ distinct bundles $b_j^1, b_j^2, \ldots, b_j^{g(j)}$. A solution to HSB is a subset $\mathcal{E}' \subseteq \mathcal{E}$ such that for every $S_j \in \mathcal{S}$ at least one bundle is covered, i.e. $b_j^l \subseteq \mathcal{E}'$. The *total cost* of the solution, denoted as $C(\mathcal{E}')$, is $\sum_{\{i|e_i \in \mathcal{E}'\}} c_i$. Notice that, the cost of an element appearing in several bundles is counted once. The objective is to find a solution with minimum total cost.

The special case of the HSB problem, in which a bundle is only an element of $\mathcal{E}$ is the classical MINIMUM HITTING SET problem[3]. It is one of the most notorious NP-hard problems and it is known to be equivalent to the classical MINIMUM SET COVER : positive and negative approximability results for the MINIMUM HITTING SET can be directly derived from the classical MINIMUM SET COVER problem [1] [4].

[3]Given a collection $\mathcal{S}$ of subsets of a finite set $\mathcal{E}$, and nonnegative costs for every element of $\mathcal{E}$, a *minimal hitting set* for $\mathcal{S}$ is a subset $\mathcal{E}' \subseteq \mathcal{E}$ such that $\mathcal{E}'$ contains at least one element from each subset in $\mathcal{S}$ and the total cost of $\mathcal{E}'$ is minimal.

[4]Recall that in the MINIMUM SET COVER, given a universe set $\mathcal{U}$, and nonnegative costs for every element of $\mathcal{U}$, a collection $\mathcal{T}$ of subsets of $\mathcal{U}$, we look for a subcollection $\mathcal{T}' \subseteq \mathcal{T}$, such that the union of the sets in $\mathcal{T}'$ is equal to $\mathcal{U}$, and $\mathcal{T}'$ is of minimal cost.

Our motivation to study the HSB problem comes not only from its own theoretical interest, but also from the fact that it models many other combinatorial optimization problems of the literature. We illustrate this fact with the *multiple-query optimization problem* (MQO for short) in database systems [10] and the MIN k-SAT PROBLEM [3].

Applications of the HSB problem

Let us first see how the MQO problem in database systems can be formulated as an HSB problem. In an instance of the MQO problem, we are given a set $Q = \{q_1, q_2, \dots, q_k\}$ of k database queries and a set $T = \{t_1, t_2, \dots, t_r\}$ of r tasks. A plan p is a subset of T and a query q_i can be solved by $n(i)$ distinct plans $P_i = \{p_i^1, p_i^2, \dots, p_i^{n(i)}\}$. Each plan is a set of elementary tasks, and each task t_j has a cost (processing time) $c_j \in \mathbb{Q}^+$. Solving the problem consists in selecting one plan per query, and the cost of a solution is the sum of the costs of the tasks involved in the selected plans (the cost of a task which belongs to at least one selected plan is counted once).

Clearly, a query of the MQO problem corresponds to a subset of $\mathcal{S}$ in the HSB problem, a plan to a bundle, and a task to an element of $\mathcal{E}$. In this context, N is the maximal number of plans per query and M, is the maximal number of queries a task can appear in. As an example, Figure 1 depicts the following instance with $Q = \{q_1, q_2, q_3\}$, $P_1 = \{p_1^1, p_1^2\}$, $P_2 = \{p_2^1, p_2^2, p_2^3\}$, $P_3 = \{p_3^1, p_3^2\}$, $p_1^1 = \{t_1, t_3, t_4\}$, $p_1^2 = \{t_1, t_2\}$, $p_2^1 = \{t_2, t_4\}$, $p_2^2 = \{t_5\}$, $p_2^3 = \{t_1, t_2, t_3\}$, $p_1^3 = \{t_1, t_3\}$, $p_3^2 = \{t_4\}$, and $c_1 = c_2 = 3$, $c_3 = c_4 = 1$, $c_5 = 2$. We have $N = M = 3$. The solution $(1\,3\,1)$[5] is feasible and its total cost is 8. Solutions $(1\,2\,1)$ and $(2\,1\,2)$ with total cost 7 are both optimal.

MQO was shown to be NP-hard in [10], and different solution methods have been proposed, including heuristics, branch and bound algorithms [10] and dynamic programming [9]. Up to now, no approximation algorithms with guaranteed performance were known for MQO.

As another application, we consider the MIN $k-$SAT problem. The input consists of a set $\mathcal{X} = \{x_1, \dots, x_t\}$ of t variables and a collection $\mathcal{C} = \{C_1, \dots, C_z\}$ of z disjunctive clauses of at most k literals (a constant ≥ 2). A literal is a variable or a negated variable in $\mathcal{X}$. A solution is a truth assignment for $\mathcal{X}$ with cost equal to the number of satisfied clauses. The objective is to find a truth assignment minimizing the number of satisfied clauses. (See in Section 4 for the reduction of MIN $k-$SAT to the HSB problem.) Kohli *et al* [7] showed that the problem is NP-hard and gave a k-approximation algorithm. Marathe and Ravi [8] improved this ratio to 2, while Bertsimas *et al* [3] showed that the problem is approximable within $2(1 - \frac{1}{2^k})$. Recently, Avidor and Zwick [2] improved the result for $k = 2$ (ratio 1.1037) and $k = 3$ (ratio 1.2136).

[5] select p_1^1, p_2^3, p_3^1

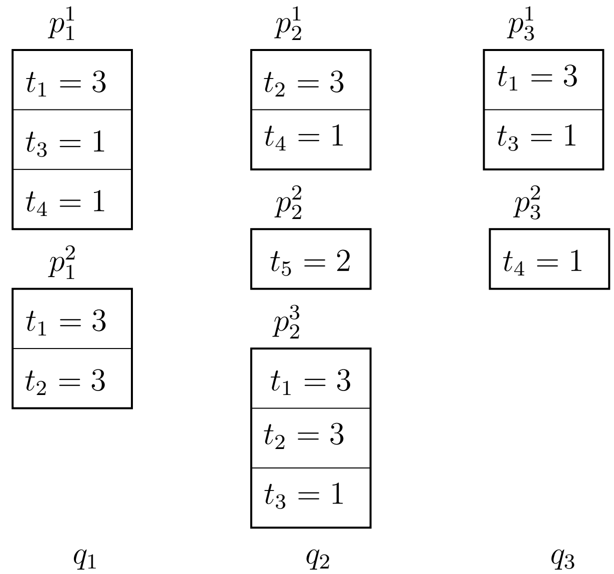

FIG. 1 – Example of the MQO problem (from [9]).

1.1 Our contribution

We give a deterministic $N(1-(1-\frac{1}{N})^M)$-approximation algorithm for the HSB problem, where N is the maximal number of bundles per set and M is the maximal number of sets an element can appear in. Our algorithm follows a rather classical scheme in the area of approximation algorithms : LP formulation, randomized rounding, derandomization. However, the analysis of the performance guarantee is quite involved. The approximation ratio is, roughly speaking, the best that we can expect for the HSB problem since we also prove that HSB cannot be approximated within $7/6-\epsilon$ when $N=2$ and $N-1-\epsilon$ when $N \geq 3$. Our algorithm is also the first approximation algorithm with guaranteed performance for the MQO problem [9, 10], while it matches the best approximation ratio for the MIN $k-$SAT problem (for general k) obtained by the algorithm of [3].

2 Inapproximability

We exploit the the fact that the MINIMUM HITTING SET problem can be formulated as a MIN VERTEX COVER in hypergraphs. In the later problem, we are given a hypergraph H and the goal is to find the smallest subset of the vertex set with non empty intersection with each hyperedge of H. Here, we are interested to the particular case of this problem where each hyperedge is composed of exactly k vertices (meaning that for the hitting set instance, each subset $S \in \mathcal{S}$ is such that $|S| = k$). We denote this case by MIN-HYPER

$k-$VERTEX COVER. When $k=2$, we get the classical MIN VERTEX COVER problem on graphs. MIN-HYPER $k-$VERTEX COVER admits a k-approximation algorithm. This result is essentially tight when $k \geq 3$ since Dinur *et al* [4] recently proved that for every $\epsilon > 0$, MIN-HYPER $k-$VERTEX COVER cannot be approximated within ratio $k-1-\epsilon$. When $k=2$, a famous result of Håstad states that MIN VERTEX COVER cannot be approximated within $7/6-\epsilon$ while a $2-\frac{2\ln\ln|V|}{\ln|V|}(1-o(1))$-approximation algorithm exists [5].

The following result can be easily obtained (the proof is omitted due to space limitations).

Theorem 1 *If there is a ρ-approximation algorithm for the* HSB *problem, then there is an approximation algorithm with the same ratio ρ for the* MIN-HYPER $k-$VERTEX COVER *problem.*

As a corollary of Theorem 1, HSB cannot be approximated within $7/6-\epsilon$ when $N=2$ and $N-1-\epsilon$ when $N \geq 3$.

3 An approximation algorithm for the HSB problem

The first natural idea is to consider a simple greedy algorithm which consists in selecting the cheapest bundle for each S_j. However, this strategy may work poorly, since the fact that some elements are shared by different bundles is not taken into account. Indeed, one can easily see that such an algorithm is M-approximate and that this ratio is reached.

Another greedy algorithm, based on the one that was originally used for the SET COVER problem [11] (using the effective cost of the subsets) does not achieve a better approximation ratio. Therefore, in what follows, we focus on LP-based algorithms.

3.1 LP-based algorithms

Solving HSB may also consist in choosing a bundle for each set of $\mathcal{S}$. This helps to formulate the problem as an integer linear program (ILP).

$$
\begin{array}{lll}
\text{minimize} & \sum_{1 \leq i \leq n} x_i\, c_i & \\
\text{subject to} & \sum_{l=1}^{g(j)} x_{j,l} \geq 1 & j = 1 \ldots m \\
& \sum_{\{l | e_i \in b_j^l\}} x_{j,l} \leq x_i & \forall (i,j) \text{ s.t. } e_i \text{ appears in a bundle of } S_j \\
& x_{j,l} \in \{0,1\} & j = 1 \ldots m \text{ and } l = 1 \ldots g(j) \\
& x_i \in \{0,1\} & i = 1 \ldots n
\end{array} \tag{1}
$$

Each bundle b_j^l is represented by a variable $x_{j,l}$ ($x_{j,l} = 1$ means b_j^l is a subset of the solution, $x_{j,l} = 0$ otherwise). Each element e_i is represented by a variable x_i ($x_i = 1$ means e_i belongs to the solution, otherwise $x_i = 0$). Among all bundles of a subset S_j, at least one is selected because of the first constraint $\sum_{l=1}^{g(j)} x_{j,l} \geq 1$. The second constraint ensures that all elements of a selected bundle appear in the solution. Since the objective function $\sum_{1 \leq j \leq r} x_j\, c_j$ has to be minimized, an element which does not belong to any selected bundle will not belong to the solution. Let LP be the linear relaxation of the ILP :

$$
\begin{array}{llll}
\text{minimize} & \sum_{1 \leq i \leq n} x_i\, c_i & & (2) \\
\text{subject to} & \sum_{l=1}^{g(j)} x_{j,l} \geq 1 & j = 1 \ldots m & (3) \\
 & \sum_{\{l | e_i \in b_j^l\}} x_{j,l} \leq x_i & \forall (i,j) \text{ s.t. } e_i \text{ appears in a bundle of } S_j & (4) \\
 & x_{j,l} \geq 0 & j = 1 \ldots m \text{ and } l = 1 \ldots g(j) & (5) \\
 & x_i \geq 0 & i = 1 \ldots n & (6)
\end{array}
$$

In the sequel, OPT and OPT_f are respectively the cost of a solution of ILP and LP (f stands for fractional). As stated before, a solution of HSB may be viewed as an m-length vector h whose jth coordinate, h_j, indicates which bundle is chosen for S_j.

We first consider a simple algorithm called D-ROUNDING : Solve LP and for $j = 1$ to m, h_j gets the value $argmax_{1 \leq l \leq g(j)}\{x_{j,l}\}$ (ties are broken arbitrarily).

Theorem 2 D-ROUNDING *is N-approximate.*

Proof. Let $\{x^*\}$ (*resp.* $\{x\}$), be an optimal assignment for ILP (*resp.* LP). One has :

$$\sum_{1 \leq i \leq n} x_i\, c_i \leq \sum_{1 \leq i \leq n} x_i^*\, c_i$$

Let $\{\tilde{x}\}$ be the solution returned by D-ROUNDING ($\tilde{x}_i = 1$ if e_i belongs to the solution and $\tilde{x}_i = 0$ otherwise). For any fixed i, if $\tilde{x}_i = 1$ then $x_i \geq 1/N$. Indeed, we take the variable whose value is the greatest (at least $1/N$ since $N = \max_j\{g(j)\}$). Then, we have $\tilde{x}_i \leq N\, x_i$ and

$$\sum_{i=1}^{n} \tilde{x}_i\, c_i \leq N \sum_{i=1}^{n} x_i\, c_i \leq N \sum_{i=1}^{n} x_i^*\, c_i$$

□

A natural idea for rounding an optimal fractional solution is to interpret fractional values of 0-1 variable as probabilities. Now, we consider an algorithm, called R-ROUNDING, of this type : Solve LP and for $j = 1$ to m, select randomly a bundle of S_j with a probability distribution $\{x_{j,1}, \ldots, x_{j,g(j)}\}$.

Theorem 3 R-ROUNDING *is* $N\big(1-(1-\frac{1}{N})^M\big)$*-approximate (in expectation).*

Before giving a proof of Theorem 3, we need some intermediate results. Let u_i be the probability of the event "e_i belongs to the solution returned by R-ROUNDING". Notice that $1-u_i \geq (1-x_i)^M$. Indeed, one has $1-u_i = \prod_{\{j|e_i \in \text{bundle of } S_j\}} \sum_{\{l'|e_i \notin b_j^{l'}\}} x_{j,l'} = \prod_{\{j|e_i \in \text{bundle of } S_j\}} (1-\sum_{\{l|e_i \in b_j^l\}} x_{j,l}) \geq \prod_{\{j|e_i \in \text{bundle of } S_j\}} (1-x_i) \geq (1-x_i)^M$. The last but one inequality comes from inequality (4), and the last inequality comes from the definition of M which is the maximal number of sets an element can appear in. Since $1-u_i \geq (1-x_i)^M$, one has $u_i \leq 1-(1-x_i)^M$. The expected cost of the solution (say $C(\mathcal{E}')$) is then bounded as follows :

$$\mathbf{E}[C(\mathcal{E}')] = \sum_{i=1}^{n} u_i\, c_i \leq \sum_{i=1}^{n} \big(1-(1-x_i)^M\big)\, c_i \tag{7}$$

In the following, we will show that :

$$\sum_{i=1}^{n} \big(1-(1-x_i)^M\big)\, c_i \leq N\big(1-(1-\frac{1}{N})^M\big) \sum_{i=1}^{n} x_i\, c_i = N\big(1-(1-\frac{1}{N})^M\big) OPT_f \tag{8}$$

where $N \geq 2$ and $M \geq 2$.[6]

We would like to prove that, for all x_i between 0 and 1, we have

$$1-(1-x_i)^M \leq N\big(1-(1-\frac{1}{N})^M\big) x_i$$

Unfortunately, the inequality is false when $0 < x_i < 1/N$. So, we show inequality (8) globally.

With simple calculus, inequality (8) becomes :

$$\sum_{i=1}^{n} x_i\, c_i \leq \big(M - N(1-(1-1/N)^M)\big)^{-1} \sum_{i=1}^{n} \big((1-x_i)^M - 1 + Mx_i\big) c_i \tag{9}$$

In the following, we prove the existence of a fractional solution $\{\tilde{x}\}$ which fulfills all the constraints of the LP, and whose total cost is an upper bound for the left part of (9) and a lower bound for the right part of (9). The proof is based on a modified version of the relaxed linear program.

[6]The problem is easy when $N = 1$ (there is only one solution) or $M = 1$ (the greedy algorithm which consists in selecting the cheapest bundle for each S_j is optimal).

Lemma 1 *There exists an assignment* $\{\tilde{x}\}$ *which fulfills all constraints of LP and the following inequalities.*

$$\sum_{i=1}^{n} x_i\, c_i \;\leq\; \sum_{i=1}^{n} \tilde{x}_i\, c_i \tag{10}$$

$$\sum_{i=1}^{n} \tilde{x}_i\, c_i \;\leq\; \big(M - N(1-(1-\frac{1}{N})^M)\big)^{-1} \sum_{i=1}^{n} \big((1-x_i)^M - 1 + Mx_i\big) c_i \tag{11}$$

Proof. Let $\{x\}$ be the values of the variables when LP is solved for a given instance of the HSB problem. Now, consider the following program called LP* for the same instance where $f(M,N,x) = \big(M - N(1-(1-1/N)^M)\big)^{-1}\big((1-x)^M - 1 + Mx\big)$.

$$\begin{array}{lll}
\text{minimize} & \sum_{1\leq i\leq n} \tilde{x}_i\, c_i & \quad (12)\\
\text{subject to} & \sum_{l=1}^{g(j)} \tilde{x}_{j,l} \geq 1 & j = 1\ldots m \quad (13)\\
 & \sum_{\{l \mid e_i \in b_j^l\}} \tilde{x}_{j,l} \leq \tilde{x}_i & \forall (i,j) \text{ s.t. } e_i \text{ appears in a bundle of } S_j \quad (14)\\
 & \tilde{x}_{j,l} \geq 0 & j = 1\ldots m \text{ and } l = 1\ldots g(j) \quad (15)\\
 & \tilde{x}_i \geq 0 & i = 1\ldots n \quad (16)\\
 & \tilde{x}_{j,l} \leq f(M,N,x_{j,l}) & j = 1\ldots m \text{ and } l = 1\ldots g(j) \quad (17)
\end{array}$$

LP* is different from LP because of the additional constraint (17). To prove that LP* always admit a solution, we need to show that (17) is not in conflict with the constraints (13) and (15).

The constraints (17) and (15) can be in conflict if $f(M,N,x) < 0$ for some x between 0 and 1. The function $f(M,N,x)$ is increasing between 0 and 1 since $f'(M,N,x) = \big(M - N(1-(1-1/N)^M)\big)^{-1}\big(M - M(1-x)^{M-1}\big) \geq 0$. Indeed, we know that $M - N(1-(1-1/N)^M) \geq 0$ since $1-(1-1/N)^M \leq M/N$. Furthermore, $M - M(1-x)^{M-1} \geq 0$ because $M \geq 1$ and $0 \leq x \leq 1$. As a consequence, $f(M,N,x) \geq 0$ when $0 \leq x \leq 1$ because $f(M,N,0) = 0$ and $f(M,N,x)$ increases. Then, the constraints (17) and (15) cannot be in conflict.

The constraints (17) and (13) can be in conflict if there exists a set of N values $\{x_t \mid 1 \leq t \leq N$ and $0 \leq x_t \leq 1\}$ such that $\sum_{t=1}^{N} x_t \geq 1$ and $\sum_{t=1}^{N} f(M,N,x_t) < 1$. The function $f(M,N,x)$ is convex when $0 \leq x \leq 1$ since $f''(M,N,x) = \big(M - N(1-(1-$

$1/N)^M)\big)^{-1}\big(M(M-1)(1-x)^{M-2}\big) \geq 0$. By the convexity, we have

$$\begin{aligned}
\frac{1}{N}\sum_{t=1}^{N} f(M,N,x_t) &\geq f(M,N,\frac{1}{N}\sum_{t=1}^{N} x_t) \geq f(M,N,\frac{1}{N}) \\
\frac{1}{N}\sum_{t=1}^{N} f(M,N,x_t) &\geq \big(M - N(1-(1-\frac{1}{N})^M)\big)^{-1}\big((1-\frac{1}{N})^M - 1 + M/N\big) \\
\sum_{t=1}^{N} f(M,N,x_t) &\geq \big(M - N(1-(1-\frac{1}{N})^M)\big)^{-1}\big(M - N(1-(1-\frac{1}{N})^M)\big) \\
\sum_{t=1}^{N} f(M,N,x_t) &\geq 1
\end{aligned}$$

Then, constraints (17) and (13) cannot be in conflict. LP* admits an assignment $\{\tilde{x}\}$ which fulfills inequality (10).

Now, we prove that the following inequality holds for each element e_i.

$$\tilde{x}_i \leq f(M,N,x_i) \tag{18}$$

Take an arbitrary element e_i. We know from LP* that for $j = 1 \ldots m$, we have

$$\tilde{x}_i \geq \sum_{\{l | e_i \in b_j^l\}} \tilde{x}_{j,l}$$

Since $\sum_{1\leq i\leq n} \tilde{x}_i\, c_i$ has to be minimized, there exists a value, say q, such that

$$\tilde{x}_i = \sum_{\{l | e_i \in b_q^l\}} \tilde{x}_{q,l}$$

By the constraint (17), we get

$$\tilde{x}_i = \sum_{\{l | e_i \in b_q^l\}} \tilde{x}_{q,l} \leq \sum_{\{l | e_i \in b_q^l\}} f(M,N,x_{q,l}) \tag{19}$$

In the Appendix, we show that

$$\sum_{\{l | e_i \in b_q^l\}} f(M,N,x_{q,l}) \leq f(M,N,\sum_{\{l | e_i \in b_q^l\}} x_{q,l}) \tag{20}$$

Using constraint (4) of the LP, we know that $\sum_{\{l | e_i \in b_q^l\}} x_{q,l}) \leq x_i$. Since f is increasing between 0 and 1, we have

$$f(M,N,\sum_{\{l | e_i \in b_q^l\}} x_{q,l}) \leq f(M,N,x_i) \tag{21}$$

Then, inequality (18) follows from (19), (20) and (21). Finally, we use (18) and the definition of f to obtain (11). □

Proof of Theorem 3.

Using Lemma 1 we know that (9) is correct and (8) follows from (9). Because of (7) and $OPT_f \leq OPT$, the result follows.

□

3.2 Derandomization

The derandomization of R-ROUNDING is done via the method of *conditional expectation* (see for example [11]). We get a deterministic algorithm called D2-ROUNDING.

Solve LP
$\mathbf{Pr}[h_j = l] = x_{j,l}$ where $j = 1 \dots m$ and $l = 1 \dots g(j)$
For $j = 1$ **to** m **Do**
  Let $l^* = \text{argmin}_{1 \leq l \leq g(j)} \mathbf{E}[C(h) \mid h_1 = l_1, \dots, h_{j-1} = l_{j-1}, h_j = l]$
  Set $l_j = l^*$

Here $E[C(h)]$ is the expected cost of a solution constructed by randomly choosing for each subset S_j a bundle (and therefore the elements inside it) according to the distribution probability given by the values $x_{j,l}$ for $l = 1 \dots g(j)$. This expected cost can be computed in polynomial time : If we note u_i the probability that element e_i belongs to the solution, recall that one has $u_i = 1 - \prod_{\{j|e_i \in \text{bundle of } S_j\}} \sum_{\{l'|e_i \notin b_j^{l'}\}} x_{j,l'}$, and we have $E[C(h)] = \sum_{i=1}^{n} u_i c_i$. In the same way, $\mathbf{E}[C(h) \mid h_1 = l_1, \dots, h_{j-1} = l_{j-1}, h_j = l]$ denotes the conditional expectation of $C(h)$ *provided* that we have chosen the bundle $b_{j'}^{l_{j'}}$ for the set $S_{j'}$ (for $1 \leq j' \leq j-1$), and bundle b_j^l for the set S_j. In the same way than before, this conditional expectation can be exactly computed in polynomial time.

Theorem 4 D2-ROUNDING *is a deterministic* $N(1-(1-\frac{1}{N})^M)$*-approximation algorithm.*

Proof.

In the following, we show that the expected cost never exceeds the original one.

Suppose we are given $l = (l_1 \dots l_{j'})$, a partial solution of the problem such that $l_1 \in \{1, \dots, g(1)\}$, $l_2 \in \{1, \dots, g(2)\}$, ..., $l_{j'} \in \{1, \dots, g(j')\}$ and $j' \in \{1, \dots, m-1\}$.

$$\begin{aligned}
&\mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}] \\
&= \sum_{l=1}^{g(j'+1)} \mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_j, h_{j'+1} = l] \, . \, \mathbf{Pr}[h_{j'+1} = l \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}] \\
&= \sum_{l=1}^{g(j'+1)} \mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}, h_{j'+1} = l] \, x_{j'+1,l}
\end{aligned}$$

If $l' = argmin_{1 \leq l \leq g(j'+1)} \mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}, h_{j'+1} = l]$ then

$$\mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}, h_{j'+1} = l'] \leq \mathbf{E}[C(h) \mid h_1 = l_1, \ldots, h_{j'} = l_{j'}]$$

At each step, the algorithm chooses a bundle (fixes its probability to 1) and the new expected cost does not exceed the previous one. Since $\mathbf{E}[C(h)] \leq N(1-(1-\frac{1}{N})^M)\, OPT$ at the beginning of the algorithm, D2-ROUNDING converges to a solution whose total cost is $N(1-(1-\frac{1}{N})^M)$-approximate. □

3.3 Integrality gap

Theorem 5 *The integrality gap of the LP is* $N(1-(1-\frac{1}{N})^M)$.

Proof. Given N and m, we can build an instance as follows.

- $\mathcal{S} = \{S_0, \ldots, S_{m-1}\}$
- $S_j = \{b_j^0, \ldots, b_j^{N-1}\}$, $j = 0, \ldots, m-1$
- $\mathcal{E} = \{e_0, \ldots, e_{N^m-1}\}$
- $c_i = 1 \; \forall e_i \in \mathcal{E}$
- Take $i \in \{0, \ldots, N^m - 1\}$ and let α be the representation of i with the numeral N-base system, i.e. $i = \sum_{j=0}^{m-1} \alpha(i,j)\, N^j$ where $\alpha(i,j) \in \{0, \ldots, N-1\}$. We set $e_i \in b_j^l$ if $\alpha(i,j) = l$.

We view solutions as vectors whose jth coordinate indicates which bundle of S_j is selected. Given a solution h, an element e_i is not selected if, for $j = 0 \ldots N-1$, we have $\alpha_i^j \neq h_j$. Then, exactly $(N-1)^m$ elements are not selected. The total cost is always $N^m - (N-1)^m$. Now consider LP. If the variable $x_{j,l}$ of each bundle b_j^l is equal to $1/N$ then the fractional cost of the solution is N^{m-1}. Indeed, an element e_i appears in exactly one bundle per S_j and the value of its variable x_i in LP is also $1/N$. As a consequence, we have $OPT_f = N^{m-1}$. Since $M = m$ in the instance, we get the following ratio

$$\frac{OPT}{OPT_f} = \frac{N^M - (N-1)^M}{N^{M-1}} = N(1-(1-\frac{1}{N})^M)$$

□

4 About MIN $k-$SAT

Theorem 6 *If there is a ρ-approximation algorithm for* HSB *then there is an approximation algorithm with the same ratio ρ for* MIN $k-$SAT.

Proof. Let A be a ρ-approximation algorithm for HSB. Take an arbitrary instance of MIN $k-$SAT and build a corresponding instance of HSB as follows. The collection $\mathcal{S}$ is made of t sets $S_1, \ldots, S_t$, one for each variable of $\mathcal{X}$. Each set S_j is composed of two bundles b_j^T and b_j^F. The set $\mathcal{E}$ contains z elements $e_1, \ldots, e_z$, one for each clause. Each element e_i has a cost $c_i = 1$. Finally, $b_j^T = \{e_i \mid C_i \text{ contains the unnegated variable } x_j\}$ and $b_j^F = \{e_i \mid C_i \text{ contains the negated variable } x_j\}$. The resulting instance of HSB is such that $N = 2$ and $M = k$.

Let τ be a truth assignment for the instance of MIN $k-$SAT with cost $C(\tau)$. One can easily derive from τ a solution h for the corresponding instance of HSB with cost $C(h) = C(\tau)$. Indeed, let h_j be T if x_j is assigned the value in τ, otherwise $h_j = F$.

Conversely, let h be a solution for the HSB instance (with $N = 2$ and $M = k$). One can easily derive a truth assignment τ for the corresponding instance of MIN $k-$SAT with cost $C(h) = C(\tau)$. Indeed, x_j gets the value *true* if $h_j = T$, otherwise x_j is assigned the value *false*.

$\square$

As a corollary of Theorem 6, MIN $k-$SAT admits a $2(1 - \frac{1}{2^k})$-approximation algorithm because D2-ROUNDING is a $N(1 - (1 - 1/N)^M)$-approximation algorithm and the reduction is such that $N = 2$ and $M = k$. This result is equivalent to the one proposed in [3].

5 Concluding remarks

Among the three deterministic approximation algorithms that we considered, D2-ROUNDING is clearly the best in terms of performance guarantee since $N(1 - (1 - 1/N)^M) < \min\{N, M\}$. Because of the integrality gap, improving this ratio with an LP-based approximation algorithm requires the use of a different (improved) formulation. An interesting direction would be to use semidefinite programming and an appropriate rounding technique as done in [5] for vertex cover in hypergraphs.

Références

[1] G. Ausiello, A. D'Atri and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1) : 136-153, 1980.

[2] A. Avidor and U. Zwick. Approximating MIN 2-SAT and MIN 3-SAT. *Theory of Computer Systems*, 38(3) : 329-345, 2005.

[3] D Bertsimas, C-P. Teo and R. Vohra. On dependent randomized rounding algorithms. *Operation Research Letters*, 24(3) : 105-114, 1999.

[4] I. Dinur, V. Guruswami, S. Khot and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. in : *Proceedings of STOC 2003*, pp 595-601, 2003.

[5] E. Halperin. Improved Approximation Algorithms for the Vertex Cover Problem in Graphs and Hypergraphs, *SIAM J. Comput.*, 31(5) :1608-1623, 2002.

[6] J. Håstad. Some optimal inapproximability results, *Journal of the ACM*, 48(4) : 798-859, 2001.

[7] R. Kohli, R. Krishnamurty and P. Mirchandani. The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7 : 275-283, 1994.

[8] M.V. Marathhe and S.S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58 : 23-29, 1996.

[9] I.H. Toroslu and A. Cosar. Dynamic programming solution for multiple query optimization problem. *Information Processing Letters*, 92(3) : 149-155, 2004.

[10] T.K. Sellis. Multiple-Query Optimization. *Transactions on Database Systems*, 13(1) : 23-52, 1988.

[11] V.V. Vazirani. Approximation Algorithms. Springer-Verlag, 2001.

Appendix

Proposition 1 *Let N and M be two positive integers, and r_1, r_2, $\dots r_N$ a set of non negatives real numbers such that $\sum_{i=1}^{N} r_i \leq 1$. Then, the following inequality holds :*

$$\sum_{i=1}^{N} f(M, N, r_i) \leq f(M, N, \sum_{i=1}^{N} r_i),$$

with $f(M, N, x) = \big(M - N(1 - (1 - \frac{1}{N})^M)\big)^{-1}\big((1-x)^M - 1 + Mx\big)$.

Proof. If a and b are two non negative reals such that $a + b \leq 1$, we observe that

$$1 - (1 - a)^M + 1 - (1 - b)^M \geq 1 - (1 - a - b)^M \tag{22}$$

Indeed, consider a probability space and two independent events, A and B, occurring from an experiment E. Let a (resp. b) be the probability of A (resp. B). Now, suppose E is repeated M times, and let A' (resp. B') be the event "A happens at least one time" (resp. "B happens at least one time"). The probability of A' (resp. B') is $1 - (1 - a)^M$ (resp. $1 - (1 - b)^M$). Let C' be the event "A or B happens at least one time". The probability of C' is $1 - (1 - a - b)^M$. We have

$$\begin{array}{rcl} Pr[A'] & = & Pr[A' \cap B'] + Pr[A' \cap \overline{B'}], \\ Pr[B'] & = & Pr[A' \cap B'] + Pr[B' \cap \overline{A'}], \\ Pr[C'] & = & Pr[A' \cap B'] + Pr[A' \cap \overline{B'}] + Pr[B' \cap \overline{A'}]. \end{array}$$

Thus, $Pr[A'] + Pr[B'] \geq Pr[C']$ and inequality (22) follows.

Then, we can apply inequality (22) $N - 1$ times to get the following inequality.

$$\sum_{i=1}^{N} \big(1 - (1 - r_i)^M\big) \geq 1 - (1 - \sum_{i=1}^{N} r_i)^M$$

It is equivalent to

$$\sum_{i=1}^{N} \big((1 - r_i)^M - 1 + Mr_i\big) \quad \leq \quad (1 - \sum_{i=1}^{N} r_i)^M - 1 + M \sum_{i=1}^{N} r_i \tag{23}$$

Let $K = \big(M - N(1 - (1 - \frac{1}{N})^M)\big)^{-1}$. We observe that $K \geq 0$ since

$$\begin{array}{rcl} (1 - 1/N)^M & \geq & 1 - M/N \\ 1 - (1 - 1/N)^M & \leq & M/N \\ N\big(1 - (1 - 1/N)^M\big) & \leq & M \\ 0 & \leq & M - N\big(1 - (1 - 1/N)^M\big) \end{array}$$

One can multiply both parts of inequality (23) by K to get the result.

□