

# On the Minimum Hitting Set of Bundles Problem

Eric Angel<sup>1</sup>, Evripidis Bampis<sup>1</sup>, and Laurent Gourvès<sup>2</sup>

1. IBISC CNRS, Université d'Evry, France  
{angel,bampis}@ibisc.fr

2. CNRS LAMSADE, Université de Paris-Dauphine, France  
laurent.gourves@lamsade.dauphine.fr

**Abstract.** We consider a natural generalization of the classical MINIMUM HITTING SET problem, the MINIMUM HITTING SET OF BUNDLES problem (MHSB) which is defined as follows. We are given a set  $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$  of  $n$  elements. Each element  $e_i$  ( $i = 1, \dots, n$ ) has a non negative cost  $c_i$ . A *bundle*  $b$  is a subset of  $\mathcal{E}$ . We are also given a collection  $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$  of  $m$  sets of bundles. More precisely, each set  $S_j$  ( $j = 1, \dots, m$ ) is composed of  $g(j)$  distinct bundles  $b_j^1, b_j^2, \dots, b_j^{g(j)}$ . A solution to MHSB 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.  $b_j^l \subseteq \mathcal{E}'$  for some  $l \in \{1, 2, \dots, g(j)\}$ . The *total cost* of the solution, denoted by  $C(\mathcal{E}')$ , is  $\sum_{\{i|e_i \in \mathcal{E}'\}} c_i$ . The goal 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 maximum number of bundles per set and  $M$  is the maximum number of sets an element can appear in. This is roughly speaking the best approximation ratio that we can obtain since, by reducing MHSB to the vertex cover problem, it implies that MHSB cannot be approximated within 1.36 when  $N = 2$  and  $N - 1 - \epsilon$  when  $N \geq 3$ . It has to be noticed that the application of our algorithm in the case of the MIN  $k$ -SAT problem matches the best known approximation ratio.

## 1 Introduction

The minimum HITTING SET OF BUNDLES problem (MHSB) is defined as follows. We are given a set  $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$  of  $n$  elements. Each element  $e_i$  ( $i = 1, \dots, n$ ) has a non negative cost  $c_i$ . A *bundle*  $b$  is a subset of  $\mathcal{E}$ . We are also given a collection  $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$  of  $m$  sets of bundles. More precisely, each set  $S_j$  ( $j = 1, \dots, m$ ) is composed of  $g(j)$  distinct bundles  $b_j^1, b_j^2, \dots, b_j^{g(j)}$ . A solution to MHSB 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.  $b_j^l \subseteq \mathcal{E}'$  for some  $l \in \{1, 2, \dots, g(j)\}$ . The *total cost* of the solution, denoted by  $C(\mathcal{E}')$ , 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 MHSB problem, in which a bundle is only an element of  $\mathcal{E}$  is the classical MINIMUM HITTING SET problem<sup>1</sup>. It is one of the most notorious NP-hard problems and it is known to be equivalent to the classical MINIMUM SET COVER problem: positive and negative approximability results for the MINIMUM HITTING SET can be directly derived from the classical MINIMUM SET COVER problem [1].<sup>2</sup>

### 1.1 Applications of the MHSB problem

Our motivation to study the MHSB 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) in database systems [10] and the MIN  $k$ -SAT problem [3].

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 MHSB problem, a plan to a bundle, and a task to an element of  $\mathcal{E}$ . In this context,  $N$  is the maximum number of plans per query and  $M$ , is the maximum number of queries a task can appear in.

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  $\geq 2$ ). 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 Section 4 for the reduction of MIN  $k$ -SAT to the MHSB problem.) Kohli *et al* [7] showed that the problem is NP-hard and gave a  $k$ -approximation algorithm. Marathe and Ravi

<sup>1</sup> 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.

<sup>2</sup> Recall that in the MINIMUM SET COVER problem, 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.

[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).

## 1.2 Contribution

We give a deterministic  $N(1 - (1 - \frac{1}{N})^M)$ -approximation algorithm for the MHSB problem, where  $N$  is the maximum number of bundles per set and  $M$  is the maximum 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 MHSB problem since, by reducing MHSB to the vertex cover problem, it implies that MHSB cannot be approximated within 1.36 when  $N = 2$  and  $N - 1 - \epsilon$  when  $N \geq 3$ .

Our algorithm matches the best approximation ratio for the MIN  $k$ -SAT problem (for general  $k$ ) obtained by the algorithm of Bertsimas *et al.* [3] and it can also be applied in the case of the MQO problem.

## 1.3 Organization of the paper

We consider the inapproximability of MHSB in Section 2 while Section 3 is devoted to its approximability. We first consider greedy strategies (proofs are deferred to the appendix) yielding to an  $M$ -approximation algorithm, followed by LP-based approximation algorithms. We first give a simple  $N$ -approximation algorithm and a randomized  $N(1 - (1 - 1/N)^M)$ -expected approximation algorithm. In Subsection 3.3, we apply a derandomization technique to derive a deterministic  $N(1 - (1 - 1/N)^M)$ -approximation algorithm. An analysis of the integrality gap conducted in Subsection 3.4 shows that the approximation result is the best we can expect. Section 4 emphasizes the link between MHSB and MIN  $k$ -SAT. We finally conclude in Section 5.

## 2 Inapproximability

We exploit the fact that the MINIMUM HITTING SET problem can be formulated as a MIN VERTEX COVER in hypergraphs. In the latter 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 in 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  $k = 2$ , the MIN VERTEX COVER problem cannot be approximated within  $10\sqrt{5} - 21 \approx 1.36$  [5] while there is a  $2 - \frac{2 \ln \ln |V|}{\ln |V|}(1 - o(1))$ -approximation algorithm [6].

From the above discussion we can deduce the following result.

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

As a corollary of Theorem 1, MHSB cannot be approximated within  $10\sqrt{5} - 21 - \epsilon$  when  $N = 2$  and  $N - 1 - \epsilon$  when  $N \geq 3$ .

### 3 Approximation algorithms

#### 3.1 Greedy algorithms

We first consider the greedy algorithm (denoted by GREEDY 1) which selects  $\operatorname{argmin}_{1 \leq l \leq g(j)} \{C(b_j^l)\}$  for every  $j$  in  $\{1, \dots, m\}$ . Actually, GREEDY 1 takes the cheapest bundle of a set without considering what is chosen for the others.

**Proposition 1.** *GREEDY 1 is an  $M$ -approximation algorithm for MHSB and the result is tight.*

We turn to a more evolved greedy algorithm which, unlike GREEDY 1, takes into account the bundles selected for other sets. The algorithm, denoted by GREEDY 2, is based on the one that was originally used for SET COVER (see [11]). Given  $\mathcal{E}' \subseteq \mathcal{E}$ , let  $B(\mathcal{E}') = |\{S_j \in \mathcal{S} \mid \exists b_j^l \subseteq \mathcal{E}'\}|$ . Actually,  $B(\mathcal{E}')$  is the number of sets in  $\mathcal{S}$  hit by  $\mathcal{E}'$ . Let  $\operatorname{Eff}(b_j^l)$  be the *effective cost* of a bundle defined as

$$\operatorname{Eff}(b_j^l) = \begin{cases} \frac{C(b_j^l \setminus \mathcal{E}')}{B(\mathcal{E}' \cup b_j^l) - B(\mathcal{E}')} & \text{if } B(\mathcal{E}' \cup b_j^l) > B(\mathcal{E}') \\ +\infty & \text{otherwise} \end{cases}$$

The algorithm uses a set  $\mathcal{E}'$  which is empty at the beginning. While  $B(\mathcal{E}') < m$ , GREEDY 2 computes the effective cost of each bundle and add to  $\mathcal{E}'$  the one which minimizes this function. Unfortunately, we can show that GREEDY 2 does not improve the performance guarantee of GREEDY 1.

**Proposition 2.** *GREEDY 2 is a  $\rho$ -approximation algorithm for MHSB such that  $\rho \geq M$ .*

#### 3.2 LP-based algorithms

Solving MHSB 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).

$$\text{minimize} \quad \sum_{1 \leq i \leq n} x_i c_i \quad (1)$$

$$\text{subject to} \quad \sum_{l=1}^{g(j)} x_{j,l} \geq 1 \quad j = 1 \dots m \quad (2)$$

$$\sum_{\{l | e_i \in b_j^l\}} x_{j,l} \leq x_i \quad \forall (i, j) \text{ s.t. } e_i \text{ appears in a bundle} \quad (3)$$

of  $S_j$

$$x_{j,l} \in \{0, 1\} \quad j = 1 \dots m \text{ and } l = 1 \dots g(j) \quad (4)$$

$$x_i \in \{0, 1\} \quad i = 1 \dots n \quad (5)$$

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  $\sum_{1 \leq j \leq m} x_j c_j$  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, i.e. replace (4) and (5) by

$$x_{j,l} \geq 0 \quad j = 1 \dots m \text{ and } l = 1 \dots g(j) \quad (6)$$

$$x_i \geq 0 \quad i = 1 \dots n \quad (7)$$

In the sequel,  $OPT$  and  $OPT_f$  are respectively the cost of a solution of ILP and LP ( $f$  stands for fractional). We will also use the following fact.

**Remark 1.** If  $\{x\}$  is an optimal solution of the LP defined by (1),(2),(3),(6),(7), we can easily prove that  $\sum_{l=1}^{g(j)} x_{j,l} = 1$  for  $j = 1, 2, \dots, m$  and that  $x_{j,l} \leq 1$  for every  $j$  and  $l$ .

Let us now consider the first simple algorithm that we call D-ROUNDING: Solve LP and for  $j = 1$  to  $m$ , select  $b_j^{h_j}$  where  $h_j = \text{argmax}_{1 \leq l \leq g(j)} \{x_{j,l}\}$  (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

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

□

Now, we consider a randomized algorithm (called R-ROUNDING) which exploits a natural idea for rounding an optimal fractional solution. It consists in interpreting fractional values of 0-1 variables as probabilities. Formally, the algorithm is as follows: Solve LP and for  $j = 1$  to  $m$ , select randomly a bundle of  $S_j$  with a probability distribution  $\{x_{j,1}, \dots, x_{j,g(j)}\}$ .

We prove that R-ROUNDING has a better approximation ratio than D-ROUNDING but, for the sake of readability, we first state two propositions and a lemma whose proofs will be given later.

**Proposition 3.** *Given two integers  $M \geq 2$ ,  $N \geq 2$  and a real  $x \in [0, 1]$ , the function  $f(M, N, x) = \frac{(1-x)^M - 1 + Mx}{M - N(1 - (1-1/N)^M)}$  is nonnegative, increasing and convex.*

**Proposition 4.** *Let  $N$ ,  $M$  and  $P$  be three positive integers such that  $P \leq N$ . Let  $r_1, r_2, \dots, r_P$  be a set of non negative reals such that  $\sum_{i=1}^P r_i \leq 1$ . The following inequality holds*

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

**Lemma 1.** *Given an instance of the MHSB problem where  $M = \max_i |\{S_j : \exists l \text{ s.t. } e_i \in b_j^l\}|$ ,  $N = \max_j \{g(j)\}$  and  $\{x\}$  is an optimal assignment for LP, there exists a feasible assignment  $\{\tilde{x}\}$  for LP which satisfies*

$$\sum_{i=1}^n \tilde{x}_i c_i \leq \sum_{i=1}^n f(M, N, x_i) c_i \quad (8)$$

We now state the main result about R-ROUNDING.

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

*Proof.* 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 (3), and the last inequality comes from the definition of  $M$ , which is the maximum 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 is then bounded as follows:

$$\mathbf{E}[C(\mathcal{E}')] = \sum_{i=1}^n u_i c_i \leq \sum_{i=1}^n (1 - (1 - x_i)^M) c_i \quad (9)$$

Using Lemma 1, we know that  $\sum_{i=1}^n x_i c_i \leq \sum_{i=1}^n \tilde{x}_i c_i$  since  $\{\tilde{x}\}$  is feasible while  $\{x\}$  is optimal. Using Lemma 1 again we obtain

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

This inequality becomes

$$\begin{aligned}
 (M - N(1 - (1 - 1/N)^M)) \sum_{i=1}^n x_i c_i &\leq \sum_{i=1}^n ((1 - x_i)^M - 1 + Mx_i) c_i \\
 (N(1 - (1 - 1/N)^M) - M) \sum_{i=1}^n x_i c_i &\geq \sum_{i=1}^n (1 - (1 - x_i)^M - Mx_i) c_i \\
 N(1 - (1 - 1/N)^M) \sum_{i=1}^n x_i c_i &\geq \sum_{i=1}^n (1 - (1 - x_i)^M) c_i
 \end{aligned}$$

Using this last inequality, inequality (9) and  $\sum_{i=1}^n x_i c_i = OPT_f \leq OPT$  we get the expected result:

$$N(1 - (1 - 1/N)^M)OPT \geq \mathbf{E}[C(\mathcal{E}')] \quad \square$$

Retrospectively, it was not possible to give a more direct proof of Theorem 3 using  $N(1 - (1 - 1/N)^M)x \geq 1 - (1 - x)^M$ , because it does not hold when  $x \in (0, \frac{1}{N})$ .

### Proof of Proposition 3

*Proof.* The function  $f(M, N, x)$  is increasing between 0 and 1 since  $f'(M, N, x) = (M - N(1 - (1 - 1/N)^M))^{-1} (M - M(1 - x)^{M-1}) \geq 0$ . Indeed, we know that  $M - N(1 - (1 - 1/N)^M) \geq 0$ .

$$\begin{aligned}
 (1 - 1/N)^M &\geq 1 - M/N \\
 1 - (1 - 1/N)^M &\leq M/N \\
 N(1 - (1 - 1/N)^M) &\leq M \\
 0 &\leq M - N(1 - (1 - 1/N)^M)
 \end{aligned} \tag{10}$$

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.

The function  $f(M, N, x)$  is convex when  $0 \leq x \leq 1$  since  $f''(M, N, x) = (M - N(1 - (1 - 1/N)^M))^{-1} (M(M - 1)(1 - x)^{M-2}) \geq 0$ . □

### Proof of Proposition 4

*Proof.* Let  $E$  be an experiment with  $N$  disjoint outcomes  $\Omega = \{O_1, O_2, \dots, O_N\}$ . Every outcome occurs with a probability  $r_i$ . Then,  $\sum_{i=1}^N r_i = 1$ . Let  $O'_i$  be the event "  $O_i$  occurs a least once when  $E$  is conducted  $M$  times ". Its probability  $\mathbf{Pr}[O'_i]$  is  $1 - (1 - r_i)^M$ . Given a nonnegative integer  $P \leq N$ , we clearly have  $\sum_{i=1}^P r_i \leq 1$ . Furthermore, the probability of the event "at least one event in

" $\{O_1, \dots, O_P\}$  occurs when  $E$  is conducted  $M$  times" is  $1 - (1 - \sum_{i=1}^P r_i)^M$ . We clearly have

$$\sum_{i=1}^P (1 - (1 - r_i)^M) \geq 1 - (1 - \sum_{i=1}^P r_i)^M, \quad (11)$$

since  $\sum_{i=1}^P \Pr[O_i] \geq \Pr[\bigcup_{i=1}^P O_i]$ . Inequality (11) gives

$$\sum_{i=1}^P ((1 - r_i)^M - 1 + Mr_i) \leq (1 - \sum_{i=1}^P r_i)^M - 1 + M \sum_{i=1}^P r_i \quad (12)$$

if we multiply by  $-1$  and add  $\sum_{i=1}^P Mr_i$  on both sides. Finally, we saw in the proof of Proposition 3 that  $M - N(1 - (1 - \frac{1}{N})^M) \geq 0$ . Then, one can divide both parts of inequality (12) by  $M - N(1 - (1 - \frac{1}{N})^M)$  to get the result.  $\square$

### Proof of Lemma 1

*Proof.* Let  $\{x\}$  be the values assigned to the variables when LP is solved. Consider the following algorithm which, given  $\{x\}$ , computes new values  $\{\tilde{x}\}$ .

```

1  For  $j = 1$  to  $m$  Do
1.1 For  $l = 1$  to  $g(j)$  Do
       $\tilde{x}_{j,l} := f(M, N, x_{j,l})$ 
      End For
      End For
2  For  $i = 1$  to  $n$  Do
       $\tilde{x}_i = \max_j \{ \sum_{\{l | e_i \in b_j^l\}} \tilde{x}_{j,l} \}$ 
      End For

```

Actually, the algorithm gives to the variable representing bundle  $b_j^l$  the value  $f(M, N, x_{j,l})$  which is always nonnegative by Proposition 3 and Remark 1. Then,  $\{\tilde{x}\}$  fulfills constraints (6) of LP.

We now show that for all  $j$  in  $\{1, \dots, m\}$ , we have  $\sum_{l=1}^{g(j)} \tilde{x}_{j,l} \geq 1$ . Let  $P = g(j')$  for a fixed  $j'$  belonging to  $\{1, \dots, m\}$ . Since  $x_{j,l}$  is optimal by Remark 1, we know that

$$\sum_{l=1}^P x_{j',l} = 1 \quad (13)$$

By the convexity of  $f$  (see Proposition 3), we have

$$\frac{1}{P} \sum_{l=1}^P f(M, N, x_{j',l}) \geq f(M, N, \frac{1}{P} \sum_{l=1}^P x_{j',l})$$

Using inequality (13), we get

$$f(M, N, \frac{1}{P} \sum_{l=1}^P x_{j',l}) = f(M, N, \frac{1}{P})$$

from which we deduce that

$$\begin{aligned} \frac{1}{P} \sum_{l=1}^P f(M, N, x_{j',l}) &\geq f(M, N, \frac{1}{P}) = \frac{(1 - \frac{1}{P})^M - 1 + M/P}{M - N(1 - (1 - \frac{1}{N})^M)} \\ \sum_{l=1}^P f(M, N, x_{j',l}) &\geq \frac{M - P(1 - (1 - \frac{1}{P})^M)}{M - N(1 - (1 - \frac{1}{N})^M)} \end{aligned} \quad (14)$$

Since  $P \leq N$ ,  $N \geq 2$  and  $M \geq 2$  we can prove the following inequality (see Proposition 5 in the Appendix).

$$M - P(1 - (1 - \frac{1}{P})^M) \geq M - N(1 - (1 - \frac{1}{N})^M) \quad (15)$$

Using (14) and (15) we deduce

$$\sum_{l=1}^P f(M, N, x_{j',l}) \geq 1 \quad (16)$$

Since no particular hypothesis was made for  $j'$ , we deduce that  $\{\tilde{x}\}$  fulfills constraints (2) of LP. Each variable  $\tilde{x}_i$  receives the value  $\max_j \{\sum_{\{l|e_i \in b_j^l\}} \tilde{x}_{j,l}\}$  at step 2 of the algorithm. Thus,  $\{\tilde{x}\}$  fulfills constraints (3) of LP. Since every  $\tilde{x}_{j,l}$  is nonnegative, we know that  $\tilde{x}_i$  is also nonnegative and  $\{\tilde{x}\}$  fulfills constraints (7) of LP. We can conclude that  $\{\tilde{x}\}$  is a feasible assignment for LP. The remaining part of the proof concerns inequality (8).

Take an element  $e_i \in \mathcal{E}$ . We know from step 2 that there is a  $q$  in  $\{1, \dots, m\}$  such that

$$\tilde{x}_i = \sum_{\{l|e_i \in b_q^l\}} \tilde{x}_{q,l} = \sum_{\{l|e_i \in b_q^l\}} f(M, N, x_{q,l}) \quad (17)$$

Using Proposition 4, we know that

$$\sum_{\{l|e_i \in b_q^l\}} f(M, N, x_{q,l}) \leq f(M, N, \sum_{\{l|e_i \in b_q^l\}} x_{q,l}) \quad (18)$$

Constraint (3) of the LP says  $\sum_{\{l|e_i \in b_q^l\}} x_{q,l} \leq x_i$ . Since  $f$  is increasing between 0 and 1, we deduce

$$f(M, N, \sum_{\{l|e_i \in b_q^l\}} x_{q,l}) \leq f(M, N, x_i) \quad (19)$$

Using inequalities (17), (18) and (19) we know that  $\tilde{x}_i \leq f(M, N, x_i)$  holds for every element  $e_i$ . We sum this inequality over all elements and obtain

$$\sum_{i=1}^n \tilde{x}_i \leq \sum_{i=1}^n f(M, N, x_i)$$

which is the expected result. □

### 3.3 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
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  $l^* = \operatorname{argmin}_{1 \leq l \leq g(j)} \mathbf{E}[C(h) \mid h_1 = l_1, \dots, h_{j-1} = l_{j-1}, h_j = l]$ 
  Set  $h_j = l^*$ 
End For

```

Here  $\mathbf{E}[C(h)]$  is the expected cost of a solution constructed by randomly choosing for each subset  $S_j$  a bundle (and therefore the elements inside) 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  $u_i = 1 - \prod_{\{j \mid e_i \in \text{bundle of } S_j\}} \sum_{\{l' \mid e_i \notin b_{j'}^{l'}\}} x_{j,l'}$ , and we have  $\mathbf{E}[C(h)] = \sum_{i=1}^n u_i c_i$ . 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  $l_1 \in \{1, \dots, g(1)\}$ ,  $l_2 \in \{1, \dots, g(2)\}$ ,  $\dots$ ,  $l_{j'} \in \{1, \dots, g(j')\}$  and  $j' \in \{1, \dots, m-1\}$ .

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

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

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

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.  $\square$

### 3.4 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.

- $\mathcal{S} = \{S_0, \dots, S_{m-1}\}$
- $S_j = \{b_j^0, \dots, b_j^{N-1}\}$ ,  $j = 0, \dots, m-1$
- $\mathcal{E} = \{e_0, \dots, e_{N^m-1}\}$
- $c_i = 1 \forall e_i \in \mathcal{E}$
- Take  $i \in \{0, \dots, N^m-1\}$  and let  $\alpha$  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  $\alpha(i, j) \in \{0, \dots, N-1\}$ . We set  $e_i \in b_j^l$  if  $\alpha(i, j) = l$ .

We view solutions as vectors whose  $j$ th coordinate indicates which bundle of  $S_j$  is selected. Given a solution  $h$ , an element  $e_i$  is not selected if, for  $j = 0, \dots, m-1$ , 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

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

$\square$

## 4 About MIN $k$ -SAT

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

*Proof.* Let  $A$  be a  $\rho$ -approximation algorithm for MHSB. Take an arbitrary instance of MIN  $k$ -SAT and build a corresponding instance of MHSB as follows. The collection  $\mathcal{S}$  is made of  $t$  sets  $S_1, \dots, S_t$ , 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  $e_1, \dots, e_z$ , 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 MHSB is such that  $N = 2$  and  $M = k$ .

Let  $\tau$  be a truth assignment for the instance of MIN  $k$ -SAT with cost  $C(\tau)$ . One can easily derive from  $\tau$  a solution  $h$  for the corresponding instance of

MHSB with cost  $C(h) = C(\tau)$ . Indeed, let  $h_j$  be  $T$  if  $x_j$  is assigned the value in  $\tau$ , otherwise  $h_j = F$ .

Conversely, let  $h$  be a solution for the MHSB instance (with  $N = 2$  and  $M = k$ ). One can easily derive a truth assignment  $\tau$  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*. □

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 by Bertsimas et al. [3].

## 5 Concluding remarks

Among the deterministic approximation algorithms that we considered, D2-ROUNDING is clearly the best in terms of performance guarantee since  $N(1 - (1 - 1/N)^M) < \min\{N, M\}$  (see inequality (10)). 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 used by Halperin [6] for vertex cover in hypergraphs.

## References

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. I. Dinur and S. Safra. The importance of being biased. in: *Proceedings of STOC 2002*, pp 33-42, 2002.
6. E. Halperin. Improved Approximation Algorithms for the Vertex Cover Problem in Graphs and Hypergraphs, *SIAM J. Comput.*, 31(5):1608-1623, 2002.
7. R. Kohli, R. Krishnamurty and P. Mirchandani. The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7: 275-283, 1994.
8. M.V. Marathe 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

### Proof of Theorem 1

*Proof.* Given any instance of the MIN-HYPER  $k$ -VERTEX COVER, we construct an instance of MHSB as follows: For each vertex  $v_i$  there is an element  $e_i$  with cost 1, for each hyperedge  $E_j = \{v_1, \dots, v_k\}$  there is a set  $S_j = \{b_j^1, \dots, b_j^k\}$  composed of  $k$  bundles such that  $b_j^i = \{v_i\}$  for  $1 \leq i \leq k$ . The result is a direct consequence of this reduction.  $\square$

### Proof of Proposition 1

*Proof.* Let  $h$  be a vector of length  $m$  such that GREEDY 1 selects  $b_j^{h_j}$  for  $S_j$ . Let  $h^*$  be a vector representing an optimal solution, i.e. selecting  $b_j^{h_j^*}$  for  $S_j$  is optimal. We have

$$\sum_{\{i|e_i \in b_j^{h_j}\}} c_i \leq \sum_{\{i|e_i \in b_j^{h_j^*}\}} c_i \quad (20)$$

for all  $j \in \{1, \dots, m\}$  because GREEDY 1 chooses the "cheapest" bundle. Summing inequality (20) for all  $j$ , we obtain

$$\sum_{j=1}^m \sum_{\{i|e_i \in b_j^{h_j}\}} c_i \leq \sum_{j=1}^m \sum_{\{i|e_i \in b_j^{h_j^*}\}} c_i \quad (21)$$

Let  $apx$  be the cost of the solution returned by GREEDY 1 while  $opt$  denotes the cost of an optimal solution. Since GREEDY 1 can select an element more than once, we have

$$apx \leq \sum_{j=1}^m \sum_{\{i|e_i \in b_j^{h_j}\}} c_i \quad (22)$$

We also have

$$Mopt \geq \sum_{j=1}^m \sum_{\{i|e_i \in b_j^{h_j^*}\}} c_i \quad (23)$$

because each element is selected at most  $M$  times by the optimal solution. Using inequalities (21), (22) and (23) we obtain  $apx \leq Mopt$ .

We build an instance with  $m + 1$  elements  $\{e_0, \dots, e_m\}$  and  $m$  sets  $\mathcal{S} = \{S_1, \dots, S_m\}$ . Each  $S_j$  has two bundles  $b_j^1 = \{e_0\}$  and  $b_j^2 = \{e_j\}$ . Every element has cost 1. The optimal solution consists in selecting  $b_j^1$  for all  $j$  while GREEDY 1 can take  $b_j^2$ . The greedy solution is  $m$ -approximate. Since  $e_0$  appears in each set, we have  $M = m$ . Thus the analysis of GREEDY 1 is tight.  $\square$

**Proof of Proposition 2**

*Proof.* We build an instance with  $2m$  elements  $\mathcal{E} = \{e_1, \dots, e_m\} \cup \{e'_1, \dots, e'_m\}$  and  $m$  sets  $\mathcal{S} = \{S_1, \dots, S_m\}$ . Each  $S_j$  has two bundles  $b_j^1 = \{e_1, \dots, e_m\} \setminus \{e_j\}$  and  $b_j^2 = \{e'_j\}$ . Every element of  $\{e_1, \dots, e_m\}$  has cost 1. Every element of  $\{e'_1, \dots, e'_m\}$  has cost  $m-1$ . The optimal solution consists in selecting  $b_j^1$  for all  $j$ . Its cost is  $|\{e_1, \dots, e_m\}| = m$ . It is not difficult to see that GREEDY 2 can select  $b_j^2$  for each  $S_j$ . The cost of the greedy solution is  $(m-1) * |\{e'_1, \dots, e'_m\}| = (m-1) * m$ . Hence it is  $(m-1)$ -approximate. Since each element in  $\{e_1, \dots, e_m\}$  appears  $m-1$  times (the others appear once), we have  $M = m - 1$ . □

**Proposition 5.** *Given  $P \leq N$ ,  $N \geq 2$  and  $M \geq 2$  we have*

$$M - P(1 - (1 - \frac{1}{P})^M) \geq M - N(1 - (1 - \frac{1}{N})^M)$$

*Proof.* Let  $f$  the function defined as  $f(x) = x(1 - (1 - 1/x)^M)$  for  $x \geq 1$ . We need to show that  $f$  is non-decreasing. One has  $f'(x) = (1 - 1/x)^{M-1}(\frac{1-M}{x} - 1) + 1$ . Let  $y = 1/x$ , one has  $f'(x) = f'(1/y) = (1 - y)^{M-1}(y - My - 1) + 1$ , for  $0 < y \leq 1$ . We want to show that  $f'(1/y) \geq 0$ , i.e.  $(1 - y)^{M-1}(1 - y + My) \leq 1$ . The proof is by induction on  $M$ . For  $M = 2$ , this inequality is true since one has  $(1 - y)(1 + y) = 1 - y^2 \leq 1$ . Let assume that  $(1 - y)^{M-1}(1 - y + My) \leq 1$ . Then  $(1 - y)^M(1 - y + (M + 1)y) = (1 - y)^{M-1}(1 - y + My)(1 - y)^{\frac{1-y+(M+1)y}{1-y+My}}$ , it is easy to see that  $(1 - y)^{\frac{1-y+(M+1)y}{1-y+My}} \leq 1$ , and the result follows. □