

Tight approximation bounds for combinatorial frugal coverage algorithms*

Ioannis Caragiannis Christos Kaklamanis Maria Kyropoulou

Department of Computer Engineering and Informatics and CTI
University of Patras, 26504 Rio, Greece

Abstract

We consider the frugal coverage problem, an interesting variation of set cover defined as follows. Instances of the problem consist of a universe of elements and a collection of sets over these elements; the objective is to compute a subcollection of sets so that the number of elements it covers plus the number of sets not chosen is maximized. The problem was introduced and studied by Huang and Svitkina [8] due to its connections to the donation center location problem. We prove that the greedy algorithm has approximation ratio at least 0.782, improving a previous bound of 0.731 in [8]. We also present a further improvement that is obtained by adding a simple corrective phase at the end of the execution of the greedy algorithm. The approximation ratio achieved in this way is at least 0.806. Finally, we consider a packing based algorithm that uses semi-local optimization, and show that its approximation ratio is not less than 0.872. Our analysis is based on the use of linear programs which capture the behavior of the algorithms in worst-case examples. The obtained bounds are proved to be tight.

1 Introduction

Set cover is among the most popular combinatorial optimization problems with many applications. In the classical version of the problem, we are given a universe of elements and a collection of sets over these elements and the objective is to compute a subcollection of sets of minimum size that covers all elements. The problem is known to be hard to approximate within sublogarithmic factors [5, 14] while the classical greedy algorithm achieves an almost tight approximation ratio of H_n , the n -th harmonic number, where n is the number of elements in the universe [11]. Several variations of the greedy algorithm have been proposed that improve this approximation bound by constant (additive) factors [1, 4, 12]. We discuss techniques that are used in such variations later in this section.

A different objective was recently considered by Huang and Svitkina [8]; they call the new combinatorial optimization problem *frugal coverage*. An instance of frugal coverage consists of a universe of elements and a collection of sets over these elements, and the objective is to compute a subcollection of sets so that the number of elements covered plus the number of sets not chosen is maximized. Without loss of generality, we can assume that each element belongs to at least one set of the input collection. So, the objective can be thought of as computing a subcollection that covers all elements so that the number of sets not chosen plus n (the number of elements in the universe) is maximized. Clearly, an optimal solution for set cover is also an optimal solution for frugal coverage. However, this does not have any direct implication to the approximation guarantee for frugal coverage algorithms.

*A preliminary version of this paper appeared in *Proceedings of the First Joint International Conference on Frontiers in Algorithmics and Algorithmic Aspects in Information and Management (FAW-AAIM 2011)*, LNCS 6681, pp. 185-195, 2011.

A nice motivation for studying frugal coverage is the problem of locating donation centers (DCL). Instances of DCL consist of a bipartite graph $G = (A \cup L, E)$. An agent, represented by a node $a \in A$, is connected through an edge $e \in E$ to any donation center $l \in L$ she would be willing to make a donation. Every agent a has a preference ranking on the corresponding centers, and every donation center l has a capacity, meaning that it can accept at most some specific number of donations. The problem is to decide which centers to open in order to maximize the number of donations, under the restriction that an agent will only donate to her highest ranked open center. Huang and Svitkina [8] present an approximation preserving reduction from frugal coverage to the special case of DCL in which each center has unit capacity and each agent has a degree bound of 2. They also prove that the greedy algorithm has approximation ratio at least 0.731 for both problems.

In this paper we study combinatorial (greedy-like) algorithms that achieve better approximation ratios for the frugal coverage problem. In the rest of this section, we independently present the different techniques used by the algorithms examined, then we informally present the algorithms themselves and explain the analysis technique used to provide the corresponding approximation bounds.

The techniques used. The *greedy algorithm* is described as follows: Starting from an empty solution, the algorithm iteratively augments the solution by a set that contains the maximum number of uncovered elements until all elements are covered. The performance of the greedy algorithm when applied on the frugal coverage problem can be further improved by adding a simple *corrective phase* at the end of the execution. This corrective phase consists of the removal of redundant sets that were used to cover two elements. In particular, we consider each set in the solution produced by the greedy algorithm, that included two uncovered elements when it was selected; if its removal does not leave an element uncovered, we simply remove this set from the solution.

Duh and Fürer introduced an interesting technique called *semi-local optimization* in [4]. Semi local optimization is a technique that extends pure local search, and was first applied on 3-set cover. The main idea is to iteratively perform improvement steps, while favoring solutions with fewer 1-sets. The term i -set, for $i = 1, \dots, k$, is used to denote a set that covered i yet uncovered elements when it was selected by the algorithm. An (s, t) -improvement step for 3-set cover consists of the insertion of up to s 3-sets and the deletion of up to t 3-sets from the current cover. Note that once the sets of size 3 have been selected, the remaining elements can be covered optimally by 2-sets and 1-sets, using a maximum matching computation. The semi-local optimization technique starts with any solution and performs (s, t) -improvement steps as long as the total number of sets in the cover is reduced, and in case of ties, prefers the cover with the minimum number of 1-sets. The analysis in [4] shows that the best choice for the parameters (s, t) is $(2, 1)$, and that, somehow surprisingly, larger values for these parameters do not yield any further improvements.

Furthermore, we make use of some known set packing heuristics. Hurkens and Schrijver [9] consider the following i -set packing algorithm: “Start with the empty packing. For any fixed s , update the solution by replacing $p \leq s$ sets by $p + 1$ sets, while you can”. They prove that for a fixed parameter $s = O(\log_i \frac{1}{\epsilon})$, an approximation ratio of $2/i - \epsilon$ is obtained, for any $\epsilon > 0$. Athanassopoulos et al. in [1] apply a combination of set packing phases and semi-local optimization on k -set cover. The idea behind the use of set packing phases in set cover is to try to cover as many elements as possible by relatively “large” sets. Their analysis is improved by Fürer and Yu [7] who apply a slightly modified packing algorithm that enhances the above i -set packing algorithm with a restriction on the number of 1-sets needed to finish the cover. They call the modified algorithm *restricted k -set packing algorithm* and provide approximation guarantees for it. The restricted k -set packing algorithm applied on a k -set cover instance starts with an arbitrary maximal collection of k -sets and repeatedly attempts to replace $p \leq s$ k -sets by $p+1$ k -sets. Such a replacement actually takes place only if the number of 1-sets computed by the semi-local optimization algorithm on the remaining instance is not more than the one computed prior to the replacement. Note that the number of 1-sets computed by the semi-local optimization lowerbounds

the number of 1-sets in any optimal solution on the corresponding instance, as shown in [4].

The algorithms. First, we present a tight analysis of the greedy algorithm, henceforth called \mathcal{A}_g . We show that its approximation ratio is exactly $18/23 \approx 0.782$, improving the previous bound in [8].

Then, we examine the performance of the greedy algorithm when a corrective phase is added at the end of its execution. Namely, when the greedy algorithm completes its execution, we consider each 2-set that was selected, and remove it from the solution if its removal does not leave an element uncovered. This algorithm is called \mathcal{A}_{gc} and it has an approximation ratio of $54/67 \approx 0.806$. A simple instance shows that this bound is tight. We remark that, even though such a corrective phase can improve the solution obtained by the greedy algorithm with respect to the standard set cover objective, it does not improve the worst-case approximation guarantee.

Furthermore, motivated by well known set packing heuristics, we consider a packing based algorithm. Our algorithm (\mathcal{A}_{gps}) combines restricted set packing phases with semi-local optimization. The algorithm works in phases starting from phase k , where k is the size of the largest set in the instance. For phase $i = k$ down to 7, \mathcal{A}_{gps} computes a maximal collection of i -sets, by greedily adding such sets to the solution. Then, for phases $i = 6, 5, 4$ of \mathcal{A}_{gps} , the restricted i -set packing algorithm is applied, and finally \mathcal{A}_{gps} applies semi-local optimization on the remaining instance. We prove that this algorithm achieves an approximation ratio of at least $75/86 \approx 0.872$.

The analysis technique. Even though the algorithms we consider are purely combinatorial, our analysis is based on the use of *linear programs*. The technique can be briefly described as follows. Given an algorithm \mathcal{A} , we define a linear program that takes a value $f \in (0, 1)$ as a parameter. This linear program witnesses the fact that the algorithm computes an at most f -approximate solution for some instance. The constraints of the LP capture the properties of the algorithm as well as the structure of the corresponding optimal solution. Then, a lower bound of ρ on the approximation ratio of algorithm \mathcal{A} follows by showing that the corresponding LP with parameter $f = \rho$ is infeasible. In order to do this, we exploit LP duality. This particular approach was recently proved to be useful for variations of set cover such as spanning star forest and color saving [2]. However, due to the different objective of frugal coverage (and, in particular, the appearance of the number of sets not chosen in the objective function), additional variables (and different constraints) have to be included in the parameterized LPs. Except from variations of set cover [1, 13], analysis of combinatorial algorithms through linear programs has also been performed in contexts such as facility location [10], wavelength management in optical networks [3], and the maximum directed cut problem [6]. The use of the LPs in these papers is slightly different than the approach followed here, since the problem objectives allow for LPs that reveal the approximation ratio of the algorithms.

Outline. We present the analysis of the greedy algorithm in Section 2 and consider its extension with an additional corrective phase in Section 3. The analysis of the set packing based algorithm with the semi-local optimization phase is presented in Section 4. We conclude in Section 5.

2 The greedy algorithm

We first consider the greedy algorithm (\mathcal{A}_g) which is described as follows. While there is a set that covers at least one uncovered element, choose the set that covers the maximum number of uncovered elements. We show that its approximation ratio for frugal coverage is exactly $18/23 \approx 0.782$. This improves the previous lower bound of 0.732 from [8].

2.1 The parameterized LP lemma

Let $f \in (0, 1)$ and consider an instance $\mathcal{I} = (\mathcal{U}, \mathcal{C})$ of the frugal coverage problem, on which \mathcal{A}_g computes an at most f -approximate solution. Without loss of generality, we assume that every element of \mathcal{U} belongs to at least one set of \mathcal{C} . We denote by \mathcal{O} the optimal solution of $(\mathcal{U}, \mathcal{C})$, i.e., the subcollection of \mathcal{C} of minimum size that covers all elements in \mathcal{U} . We assume that any set of \mathcal{C} belongs either to the optimal solution or is chosen by \mathcal{A}_g ; if this is not the case for the original instance, we can easily transform it to one that satisfies this assumption so that the solution computed by \mathcal{A}_g is at most f -approximate.

The algorithm can be thought of as running in phases, starting from phase k , where k is the size of the largest set in the instance. In each phase i , for $i = k, \dots, 1$, the algorithm chooses a maximal collection of sets, each containing i yet uncovered elements. Let $(\mathcal{U}_i, \mathcal{C}_i)$, for $i = k, \dots, 1$, denote the corresponding instance which remains to be solved just before entering phase i of the algorithm. Naturally, \mathcal{U}_i consists of the elements in \mathcal{U} that have not been covered in previous phases and \mathcal{C}_i consists of the sets of \mathcal{C} which contain at least one such element. We denote by \mathcal{O}_i the sets in \mathcal{O} that also belong to \mathcal{C}_i . We consider an assignment of all elements to the sets of \mathcal{O} so that each element is assigned to exactly one among the sets in \mathcal{O} that contains it (if more than one). A set in \mathcal{O}_i is called an (i, j) -set if exactly j elements that have not been covered until the beginning of phase i are assigned to it. For the phase i of the algorithm with $i = 4, \dots, 1$, we denote by $\alpha_{i,j}$ the ratio of the number of (i, j) -sets in \mathcal{O}_i over the number $|\mathcal{O}|$ of optimal sets. Furthermore, let x_5 be the ratio of the number of sets in \mathcal{O} that are chosen by \mathcal{A}_g at the phases $k, k-1, \dots, 5$ over $|\mathcal{O}|$. Also, let x_i , for $i = 4, \dots, 1$, denote the ratio of the number of sets in \mathcal{O} that are also selected by \mathcal{A}_g in phase i , over $|\mathcal{O}|$.

By definition, it holds that $|\mathcal{O}_i| = |\mathcal{O}| \sum_{j=1}^i \alpha_{i,j}$ and $|\mathcal{O}_i| \leq |\mathcal{O}| - |\mathcal{O}| \sum_{j=i+1}^5 x_j$, for $i = 1, \dots, 4$. Combining these expressions, we get that

$$\sum_{j=1}^i \alpha_{i,j} + \sum_{j=i+1}^5 x_j \leq 1. \quad (1)$$

We denote by T the ratio $|\mathcal{U}|/|\mathcal{O}|$. Our definitions imply that $|\mathcal{U}_i| = |\mathcal{O}| \sum_{j=1}^i j \alpha_{i,j}$ for $i = 1, \dots, 4$. By the definitions of T and x_5 , we have

$$T \geq \sum_{j=1}^4 j \alpha_{4,j} + 5x_5. \quad (2)$$

Clearly, for $i = 1, \dots, 4$, it holds that $|\mathcal{U}_i \setminus \mathcal{U}_{i-1}| = |\mathcal{U}_i| - |\mathcal{U}_{i-1}| \geq ix_i \cdot |\mathcal{O}|$. We have

$$\sum_{j=1}^i j \alpha_{i,j} \geq \sum_{j=1}^{i-1} j \alpha_{i-1,j} + ix_i. \quad (3)$$

Now, consider phase i of the algorithm, for $i = 4, 3, 2$, and a set chosen by \mathcal{A}_g during this phase. The i newly covered elements of this set are assigned to at most i sets in \mathcal{O}_i . Furthermore, since \mathcal{A}_g selects a maximal collection of sets in phase i , we know that any set of \mathcal{O}_i with i assigned elements intersects some of the sets selected by \mathcal{A}_g during this phase. This means that the number $|\mathcal{U}_i \setminus \mathcal{U}_{i-1}|/i$ of sets selected by \mathcal{A}_g during phase i is at least $\alpha_{i,i}|\mathcal{O}|/i$. We obtain that

$$\sum_{j=1}^i j \alpha_{i,j} - \sum_{j=1}^{i-1} j \alpha_{i-1,j} \geq \alpha_{i,i}. \quad (4)$$

Now, let \mathcal{S}_G denote the ratio of the number of sets chosen by \mathcal{A}_g over $|\mathcal{O}|$. Clearly, \mathcal{A}_g selects at most $|\mathcal{U} \setminus \mathcal{U}_4|/5$ sets in phases $k, k-1, \dots, 5$, and exactly $|\mathcal{U}_i \setminus \mathcal{U}_{i-1}|/i$ sets in phase i , for $i = 4, \dots, 1$.

We have that

$$\begin{aligned}
\mathcal{S}_G &\leq \left(\frac{1}{5}|\mathcal{U} \setminus \mathcal{U}_4| + \sum_{i=1}^4 \frac{1}{i}|\mathcal{U}_i \setminus \mathcal{U}_{i-1}| \right) / |\mathcal{O}| \\
&= \frac{1}{5} \left(T - \sum_{j=1}^4 j\alpha_{4,j} \right) + \sum_{i=1}^4 \frac{1}{i} \left(\sum_{j=1}^i j\alpha_{i,j} - \sum_{j=1}^{i-1} j\alpha_{i-1,j} \right) \\
&= \frac{1}{5}T + \sum_{i=1}^4 \frac{1}{i(i+1)} \sum_{j=1}^i j\alpha_{i,j}.
\end{aligned}$$

Let $\text{OPT}(\mathcal{I})$ denote the benefit of the optimal frugal coverage of \mathcal{I} . This naturally corresponds to solution \mathcal{O} for $(\mathcal{U}, \mathcal{C})$. It holds that

$$\text{OPT}(\mathcal{I}) = \left(T + \mathcal{S}_G - \sum_{i=1}^5 x_i \right) |\mathcal{O}|.$$

Furthermore, let $\mathcal{A}_g(\mathcal{I})$ denote the profit \mathcal{A}_g obtains on \mathcal{I} . We have that

$$\mathcal{A}_g(\mathcal{I}) = \left(T + 1 - \sum_{i=1}^5 x_i \right) |\mathcal{O}|.$$

The assumption that the solution obtained by \mathcal{A}_g is at most f -approximate, i.e., $\mathcal{A}_g(\mathcal{I}) \leq f \cdot \text{OPT}(\mathcal{I})$, implies that

$$T + 1 - \sum_{i=1}^5 x_i \leq f \left(\frac{6}{5}T + \sum_{i=1}^4 \frac{1}{i(i+1)} \sum_{j=1}^i j\alpha_{i,j} - \sum_{i=1}^5 x_i \right). \quad (5)$$

By expressing inequalities (1)-(5) in standard form, we obtain our parameterized LP lemma.

Lemma 1 *If there exists an instance \mathcal{I} of the frugal coverage problem for which the greedy algorithm computes a solution of benefit $\mathcal{A}_g(\mathcal{I}) \leq f \cdot \text{OPT}(\mathcal{I})$ for some $f \in [0, 1]$, then the following linear program $LP(f)$ has a feasible solution.*

$$\begin{aligned}
&\sum_{j=1}^i \alpha_{i,j} + \sum_{j=i+1}^5 x_j \leq 1, \quad \text{for } i = 1, \dots, 4 \\
&-(i-1)\alpha_{i,i} - \sum_{j=1}^{i-1} j\alpha_{i,j} + \sum_{j=1}^{i-1} j\alpha_{i-1,j} \leq 0, \quad \text{for } i = 2, 3, 4 \\
&-\sum_{j=1}^i j\alpha_{i,j} + \sum_{j=1}^{i-1} j\alpha_{i-1,j} + ix_i \leq 0, \quad \text{for } i = 1, \dots, 4 \\
&-T + \sum_{j=1}^4 j\alpha_{4,j} + 5x_5 \leq 0 \\
&(1 - \frac{6f}{5})T - \sum_{i=1}^4 \sum_{j=1}^i \frac{fj}{i(i+1)} \alpha_{i,j} - \sum_{i=1}^5 (1-f)x_i \leq -1 \\
&\alpha_{i,j} \geq 0, \quad \text{for } i = 1, \dots, 4 \text{ and } j = 1, \dots, i \\
&x_j \geq 0, \quad \text{for } j = 1, \dots, 5 \\
&T \geq 0
\end{aligned}$$

The proof of the approximation bound is based on the following lemma.

Lemma 2 *For every $f < 18/23$, $LP(f)$ has no feasible solution.*

Proof. We can assume that $LP(f)$ is a maximization linear program with objective 0. By duality, if it were feasible, then the optimal objective value of the dual minimization linear program should be 0 as well. We show that this is not the case and that the dual has a solution with strictly negative objective value. This implies the lemma.

In the dual LP, we use the thirteen variables $\beta_1, \beta_2, \beta_3, \beta_4, \gamma_2, \gamma_3, \gamma_4, \delta_1, \delta_2, \delta_3, \delta_4, \eta$ and ζ corresponding to the constraints of $LP(f)$. Variables β_i correspond to the first set of constraints (inequality (1)) of $LP(f)$, variables γ_i correspond to the second set of constraints (inequality (4)), δ_i correspond to the third set of constraints (inequality (3)), η corresponds to the next constraint (inequality (2)), and ζ corresponds to the last constraint (inequality (5)). So, the dual of $LP(f)$ is depicted in Table 1.

$$\begin{array}{ll}
\min & \beta_1 + \beta_2 + \beta_3 + \beta_4 - \zeta \\
\\
\text{s.t.} & \beta_1 + \gamma_2 - \delta_1 + \delta_2 - \zeta f/2 \geq 0 & \beta_4 - 3\gamma_4 - 3\delta_4 + 3\eta - \zeta 3f/20 \geq 0 \\
& \beta_2 - \gamma_2 + \gamma_3 - \delta_2 + \delta_3 - \zeta f/6 \geq 0 & \beta_4 - 3\gamma_4 - 4\delta_4 + 4\eta - \zeta f/5 \geq 0 \\
& \beta_2 - \gamma_2 + 2\gamma_3 - 2\delta_2 + 2\delta_3 - \zeta f/3 \geq 0 & \delta_1 - (1-f)\zeta \geq 0 \\
& \beta_3 - \gamma_3 + \gamma_4 - \delta_3 + \delta_4 - \zeta f/12 \geq 0 & \beta_1 + 2\delta_2 - (1-f)\zeta \geq 0 \\
& \beta_3 - 2\gamma_3 + 2\gamma_4 - 2\delta_3 + 2\delta_4 - \zeta f/6 \geq 0 & \beta_1 + \beta_2 + 3\delta_3 - (1-f)\zeta \geq 0 \\
& \beta_3 - 2\gamma_3 + 3\gamma_4 - 3\delta_3 + 3\delta_4 - \zeta f/4 \geq 0 & \beta_1 + \beta_2 + \beta_3 + 4\delta_4 - (1-f)\zeta \geq 0 \\
& \beta_4 - \gamma_4 - \delta_4 + \eta - \zeta f/20 \geq 0 & \beta_1 + \beta_2 + \beta_3 + \beta_4 + 5\eta - (1-f)\zeta \geq 0 \\
& \beta_4 - 2\gamma_4 - 2\delta_4 + 2\eta - \zeta f/10 \geq 0 & -\eta + (1-6f/5)\zeta \geq 0 \\
& \beta_i, \delta_i \geq 0, \quad \text{for } i = 1, \dots, 4 \\
& \gamma_i \geq 0, \quad \text{for } i = 2, 3, 4 \\
& \zeta, \eta \geq 0
\end{array}$$

Table 1: The dual of $LP(f)$ in the proof of Lemma 2.

The solution $\beta_1 = f/2 + 1/46, \beta_2 = f/4, \beta_3 = f/3, \beta_4 = 15/46 - f/4, \gamma_2 = f/4, \gamma_3 = f/6, \gamma_4 = 5/46 - f/12, \delta_1 = 1 - f, \delta_2 = \delta_3 = 45/46 - 5f/4, \delta_4 = 1 - 5f/4, \eta = 1 - 6f/5$ and $\zeta = 1$ satisfies all the constraints. Observe that $\beta_1 + \beta_2 + \beta_3 + \beta_4 - \zeta = \frac{5f}{6} - \frac{15}{23}$, which implies that the objective value of the dual program is strictly negative since $f < \frac{18}{23}$. The lemma follows. ■

Theorem 3 *The approximation ratio of the greedy algorithm is at least $18/23$.*

Proof. By Lemmas 1 and 2, we have that for any $f < 18/23$ and for any instance \mathcal{I} of the frugal coverage problem, the greedy algorithm computes a solution of benefit $\mathcal{A}_g(\mathcal{I}) > f \cdot \text{OPT}(\mathcal{I})$. Hence, its approximation ratio is at least $18/23$. ■

2.2 The upper bound

The instance depicted in Figure 1 proves that our analysis is tight. It consists of 18 elements and 11 sets. The greedy algorithm starts by choosing the 2 horizontal sets of size 3, then it chooses the 3 horizontal sets of size 2, and, finally, it chooses all the vertical sets in order to cover the last 6 elements. Thus, \mathcal{A}_g uses all 11 sets in order to cover the 18 elements for a profit of 18. The optimal solution consists of the 6 vertical sets of size 3 and has a profit of 23 since it covers the 18 elements without using 5 out of the 11 sets. The upper bound follows.

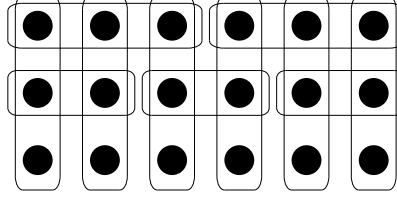


Figure 1: The tight upper bound for \mathcal{A}_g .

3 Adding a corrective phase

In this section we show that an improvement in the performance of the greedy algorithm is obtained by adding a simple corrective phase at the end of its execution. We call the new algorithm \mathcal{A}_{gc} and prove that its approximation ratio is exactly $54/67 \approx 0.8059$ for frugal coverage. The definition of the corrective phase is very simple. After the execution of \mathcal{A}_g , we examine every set s that was chosen during phase 2; if by removing s we still have a cover of the elements, we simply remove s from the solution.

3.1 The parameterized LP lemma

We slightly modify the parameterized LP for \mathcal{A}_g in order to capture the behavior of the corrective phase. We will need some additional definitions. We denote by \mathcal{G}_2 and \mathcal{G}_1 the sets included by the algorithm in phases 2 and 1, respectively. A $(2, 1)$ -set of \mathcal{O}_2 is of type Γ if its uncovered element is included in a set of \mathcal{G}_2 . A $(2, 2)$ -set of \mathcal{O}_2 is of type A if both of its uncovered elements are included in sets of \mathcal{G}_2 and is of type B otherwise. A set of \mathcal{G}_2 is of type AA if it does not belong to \mathcal{O}_2 and its newly covered elements (during phase 2) belong to $(2, 2)$ -sets of \mathcal{O}_2 of type A and is of type $kl \in \{AB, A\Gamma, BB, B\Gamma\}$ if its newly covered elements belong to two sets of \mathcal{O}_2 of types k and l , respectively. A set of \mathcal{G}_1 is of type Δ if its newly covered element belongs to a $(2, 1)$ -set of \mathcal{O}_2 . We introduce the variables $t_{AA}, t_{AB}, t_{A\Gamma}, t_{BB}, t_{B\Gamma}, t_{\Gamma\Gamma}$, and t_{Δ} to denote the ratio of the number of sets of types $AA, AB, A\Gamma, BB, B\Gamma, \Gamma\Gamma$, and Δ over $|\mathcal{O}|$, respectively. We also use variable d to denote the ratio of the number of sets removed during the corrective phase over $|\mathcal{O}|$. The next lemma provides a lower bound on d , which we will use as a constraint in the parameterized LP for \mathcal{A}_{gc} .

Lemma 4 $x_1 - t_{AB} - t_{BB} - t_{B\Gamma} - t_{\Delta} - d \leq 0$.

Proof. We partition $(2, 2)$ -sets of \mathcal{O}_2 of type B into two subtypes. Such a $(2, 2)$ -set s is of type \hat{B} if one of its elements is included in a set of \mathcal{G}_1 that also belongs to \mathcal{O}_2 and is of type \bar{B} otherwise. Now, we extend the notation t_k to denote the ratio of the number of sets of \mathcal{G}_2 of type $k \in \{A\hat{B}, A\bar{B}, \hat{B}\hat{B}, \hat{B}\bar{B}, \bar{B}\bar{B}, \hat{B}\Gamma, \bar{B}\Gamma\}$ over $|\mathcal{O}|$.

Observe that for each set s of \mathcal{O}_2 of type \hat{B} , there exists a set in $\mathcal{O}_1 \cap \mathcal{G}_1$ (the one containing the uncovered element of s at the beginning of phase 1) and any other set of $\mathcal{O}_1 \cap \mathcal{G}_1$ is of type Δ . Hence, $x_1 \leq t_{A\hat{B}} + 2t_{\hat{B}\hat{B}} + t_{\hat{B}\bar{B}} + t_{\hat{B}\Gamma} + t_{\Delta}$. Now, observe that the number of sets that will be removed during the corrective phase include those of type $\hat{B}\hat{B}$. Hence,

$$d \geq t_{\hat{B}\hat{B}} \geq x_1 - t_{A\hat{B}} - t_{\hat{B}\hat{B}} - t_{\hat{B}\bar{B}} - t_{\hat{B}\Gamma} - t_{\Delta} \geq x_1 - t_{AB} - t_{BB} - t_{B\Gamma} - t_{\Delta},$$

as desired. ■

Variables $a_{1,1}$, $a_{2,1}$, and $a_{2,2}$ will not be explicitly used; observe that we can replace them as follows:

$$\begin{aligned}\alpha_{1,1} &= t_{AB} + 2t_{BB} + t_{B\Gamma} + t_{\Delta} \\ \alpha_{2,1} &= t_{A\Gamma} + t_{B\Gamma} + 2t_{\Gamma\Gamma} + t_{\Delta}, \text{ and} \\ \alpha_{2,2} &= t_{AA} + \frac{3}{2}t_{AB} + \frac{1}{2}t_{A\Gamma} + 2t_{BB} + t_{B\Gamma} + x_2.\end{aligned}$$

The profit $\mathcal{A}_{gc}(\mathcal{I})$ is now $d|\mathcal{O}|$ more than $\mathcal{A}_g(\mathcal{I})$, i.e.,

$$\mathcal{A}_{gc}(\mathcal{I}) = \left(T + 1 - \sum_{i=1}^5 x_i + d \right) |\mathcal{O}|,$$

while $\text{OPT}(\mathcal{I})$ is the same as in the previous section. In this way, we have obtained the following parameterized LP lemma for \mathcal{A}_{gc} .

Lemma 5 *If there exists an instance \mathcal{I} of the frugal coverage problem for which \mathcal{A}_{gc} computes a solution of benefit $\mathcal{A}_{gc}(\mathcal{I}) \leq f \cdot \text{OPT}(\mathcal{I})$ for some $f \in [0, 1]$, then the linear program $\text{LP}_{gc}(f)$ has a feasible solution.*

$$\begin{aligned}t_{AB} + 2t_{BB} + t_{B\Gamma} + t_{\Delta} + \sum_{j=2}^5 x_j &\leq 1 \\ t_{AA} + \frac{3}{2}t_{AB} + \frac{3}{2}t_{A\Gamma} + 2t_{BB} + 2t_{B\Gamma} + 2t_{\Gamma\Gamma} + t_{\Delta} + x_2 + \sum_{j=3}^5 x_j &\leq 1 \\ \sum_{j=1}^i \alpha_{i,j} + \sum_{j=i+1}^5 x_j &\leq 1, \quad \text{for } i = 3, 4 \\ -2\alpha_{3,3} - \sum_{j=1}^2 j\alpha_{3,j} + 2t_{AA} + 3t_{AB} + 2t_{A\Gamma} + 4t_{BB} + 3t_{B\Gamma} + 2t_{\Gamma\Gamma} + t_{\Delta} + 2x_2 &\leq 0 \\ -3\alpha_{4,4} - \sum_{j=1}^3 j\alpha_{4,j} + \sum_{j=1}^3 j\alpha_{3,j} &\leq 0 \\ -\sum_{j=1}^3 j\alpha_{3,j} + 2t_{AA} + 3t_{AB} + 2t_{A\Gamma} + 4t_{BB} + 3t_{B\Gamma} + 2t_{\Gamma\Gamma} + t_{\Delta} + 2x_2 + 3x_3 &\leq 0 \\ -\sum_{j=1}^4 j\alpha_{4,j} + \sum_{j=1}^3 j\alpha_{3,j} + 4x_4 &\leq 0 \\ -T + \sum_{j=1}^4 j\alpha_{4,j} + 5x_5 &\leq 0 \\ x_1 - t_{AB} - t_{BB} - t_{B\Gamma} - t_{\Delta} - d &\leq 0\end{aligned}$$

$$\begin{aligned}
& (1 - \frac{6f}{5})T - \frac{f}{3}t_{AA} - ft_{AB} - \frac{f}{3}t_{A\Gamma} - \frac{5f}{3}t_{BB} - ft_{B\Gamma} - \frac{f}{3}t_{\Gamma\Gamma} - \frac{2f}{3}t_{\Delta} \\
& - \sum_{i=3}^4 \sum_{j=1}^i \frac{fj}{i(i+1)}\alpha_{i,j} - \sum_{i=3}^5 (1-f)x_i - (1-f)x_1 - (1 - \frac{2f}{3})x_2 + d \leq -1 \\
& \alpha_{i,j} \geq 0, \quad \text{for } i = 3, 4 \text{ and } j = 1, \dots, i \\
& x_j \geq 0, \quad \text{for } j = 1, \dots, 5 \\
& t_{AA}, t_{AB}, t_{A\Gamma}, t_{BB}, t_{B\Gamma}, t_{\Gamma\Gamma}, t_{\Delta} \geq 0 \\
& T \geq 0
\end{aligned}$$

The proof of the approximation bound is based on the following lemma.

Lemma 6 For every $f < 54/67$, $LP_{gc}(f)$ has no feasible solution.

Proof. Similarly to the proof of Lemma 2, we assume that $LP_{gc}(f)$ is a maximization linear program with objective 0. We show that its dual has a solution with strictly negative objective value when $f < 54/67$. This implies the lemma.

To construct the dual of $LP_{gc}(f)$ we use the eleven variables $\beta_1, \beta_2, \beta_3, \beta_4, \gamma_3, \gamma_4, \delta_3, \delta_4, \eta, \mu$ and ζ corresponding to the constraints of $LP_{gc}(f)$. Variables β_i correspond to the constraints in the first three lines of $LP_{gc}(f)$, variables γ_i correspond to the constraints in the fourth and fifth line, variables δ_i correspond to the constraints in the sixth and seventh line, and $\eta, \mu,$ and ζ correspond to the three last constraints of $LP_{gc}(f)$, respectively. So, the dual of $LP_{gc}(f)$ is depicted in Table 2.

$$\begin{array}{ll}
min & \beta_1 + \beta_2 + \beta_3 + \beta_4 - \zeta \\
s.t. & \beta_2 + 2\gamma_3 + 2\delta_3 - \frac{f}{3}\zeta \geq 0 & \beta_4 - \gamma_4 - \delta_4 + \eta - \frac{f}{20}\zeta \geq 0 \\
& \beta_1 + \frac{3}{2}\beta_2 + 3\gamma_3 + 3\delta_3 - \mu - f\zeta \geq 0 & \beta_4 - 2\gamma_4 - 2\delta_4 + 2\eta - \frac{f}{10}\zeta \geq 0 \\
& \frac{3}{2}\beta_2 + 2\gamma_3 + 2\delta_3 - \frac{f}{3}\zeta \geq 0 & \beta_4 - 3\gamma_4 - 3\delta_4 + 3\eta - \frac{3f}{20}\zeta \geq 0 \\
& 2\beta_1 + 2\beta_2 + 4\gamma_3 + 4\delta_3 - \mu - \frac{5f}{3}\zeta \geq 0 & \beta_4 - 3\gamma_4 - 4\delta_4 + 4\eta - \frac{f}{5}\zeta \geq 0 \\
& \beta_1 + 2\beta_2 + 3\gamma_3 + 3\delta_3 - \mu - f\zeta \geq 0 & \mu - (1-f)\zeta \geq 0 \\
& 2\beta_2 + 2\gamma_3 + 2\delta_3 - \frac{f}{3}\zeta \geq 0 & \beta_1 + \beta_2 + 2\gamma_3 + 2\delta_3 - (1 - \frac{2f}{3})\zeta \geq 0 \\
& \beta_1 + \beta_2 + \gamma_3 + \delta_3 - \mu - \frac{2f}{3}\zeta \geq 0 & \beta_1 + \beta_2 + 3\delta_3 - (1-f)\zeta \geq 0 \\
& \beta_3 - \gamma_3 + \gamma_4 - \delta_3 + \delta_4 - \frac{f}{12}\zeta \geq 0 & \beta_1 + \beta_2 + \beta_3 + 4\delta_4 - (1-f)\zeta \geq 0 \\
& \beta_3 - 2\gamma_3 + 2\gamma_4 - 2\delta_3 + 2\delta_4 - \frac{f}{6}\zeta \geq 0 & \beta_1 + \beta_2 + \beta_3 + \beta_4 + 5\eta - (1-f)\zeta \geq 0 \\
& \beta_3 - 2\gamma_3 + 3\gamma_4 - 3\delta_3 + 3\delta_4 - \frac{f}{4}\zeta \geq 0 & -\eta + (1 - 6f/5)\zeta \geq 0 \\
& \beta_i \geq 0, \quad \text{for } i = 1, \dots, 4 & -\mu + \zeta \geq 0 \\
& \gamma_i, \delta_i \geq 0, \quad \text{for } i = 3, 4 \\
& \zeta, \mu, \eta \geq 0
\end{array}$$

Table 2: The dual of $LP_{gc}(f)$ in the proof of Lemma 6.

The solution $\beta_1 = 41/134, \beta_2 = f/3 + 13/67, \beta_3 = 0, \beta_4 = 31f/108, \gamma_3 = 0, \gamma_4 = f/12, \delta_3 = \delta_4 = 1 - 67f/54, \eta = 1 - 6f/5, \zeta = 1$ and $\mu = 1 - f$ satisfies all the constraints. Observe that $\beta_1 + \beta_2 + \beta_3 + \beta_4 - \zeta = \frac{67f}{108} - \frac{1}{2}$, which implies that the objective value of the dual program is strictly negative for $f < \frac{54}{67}$. The lemma follows. \blacksquare

The next statement follows by Lemmas 5 and 6.

Theorem 7 The approximation ratio of \mathcal{A}_{gc} is at least $54/67$.

3.2 The upper bound

The instance which yields the tight upper bound for \mathcal{A}_{gc} consists of 48 elements and 31 sets; see Figure 2. There, the sets selected by \mathcal{A}_{gc} are shown, while the optimal solution consists of the 12 vertical disjoint sets of size 4 (note that only half of them are shown in the figure). Clearly, the profit obtained by \mathcal{A}_{gc} is $48 + 6 = 54$, while the profit of the optimal solution is at least $48 + 19 = 67$, and the upper bound follows. We remark that the corrective phase does not remove any set.

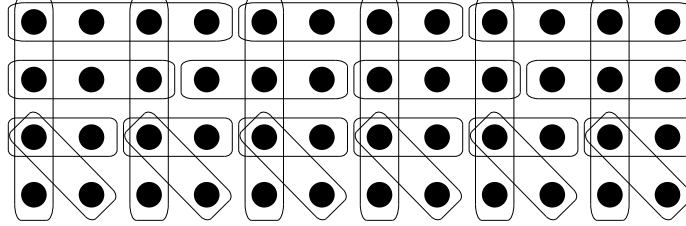


Figure 2: The tight upper bound for \mathcal{A}_{gc} .

4 Greedy with set packing phases and semi-local optimization

In this section we analyze the performance of the greedy algorithm when it is combined with restricted set packing phases as well as a semi-local optimization phase. We denote this algorithm by \mathcal{A}_{gps} and start by presenting its execution. For $i = k, \dots, 7$, the algorithm first greedily includes disjoint i -sets in the solution until no other i -set can be included. Then, the algorithm executes a restricted set packing phase in order to pick a maximal collection of disjoint 6-sets while it guarantees that the number of 1-sets in the final solution is not larger than the number of 1-sets in the optimal solution of the 6-set packing instance at the beginning of this restricted phase. Then, it applies a similar restricted set packing phase for 5-sets and 4-sets. After this phase, a $(\mathcal{U}_3, \mathcal{C}_3)$ instance remains to be solved in order to complete the solution; the algorithm applies a semi-local optimization phase on it.

4.1 The parameterized LP lemma

First we give a more formal description of the algorithm.

$\mathcal{A}_{gps}(\mathcal{U}, \mathcal{C}) :$

Greedy phases: For $i = k$ down to 7, do:

Choose a maximal collection of disjoint i -sets.

Restricted Set Packing phases: For $i = 6$ down to 4, do:

Select a maximal collection of disjoint i -sets using a ρ_i -approximation restricted i -set packing algorithm. ($\rho_i = \frac{2}{i} - \epsilon$, for $i = 6, 5$, and $\rho_4 = \frac{7}{16}$)

Semi-local optimization phase: Run the semi-local optimization algorithm on the remaining $(\mathcal{U}_3, \mathcal{C}_3)$ instance.

In [7], Fürer and Yu guarantee the existence of the set packing algorithms with the specific approximation ratios, as stated in the theorem.

Theorem 8 (Fürer and Yu [7]) *There exists a restricted 4-set packing algorithm which has approximation ratio $\frac{7}{16}$. For all $\epsilon > 0$ and for any $i \geq 5$ there exists a restricted i -set packing algorithm which has approximation ratio $\frac{2}{i} - \epsilon$.*

Next, we present the analysis of the \mathcal{A}_{gps} algorithm. Having as a starting point the analysis of the greedy algorithm in Section 2, we modify it accordingly in order to capture the extra constraints imposed by the restricted set packing phases as well as the semi-local optimization phase.

For the analysis of \mathcal{A}_{gps} we use variables $\alpha_{i,j}$, for $i = 2, \dots, 6$ and $j = 1, \dots, i$, variables x_j , for $j = 1, \dots, 7$, and variable T following their definition in Section 2. Our algorithm uses restricted set packing phases in an effort to maximize the number of disjoint sets selected (and hence, elements covered) in phases 6, 5, and 4. The approximation guarantees in Theorem 8 imply the following inequalities, which dominate the corresponding ones in equation (4).

$$\frac{\sum_{j=1}^4 j\alpha_{4,j} - \sum_{j=1}^3 j\alpha_{3,j}}{4} \geq \frac{7}{16}\alpha_{4,4}, \quad \text{and}$$

$$\frac{\sum_{j=1}^i j\alpha_{i,j} - \sum_{j=1}^{i-1} j\alpha_{i-1,j}}{i} \geq \left(\frac{2}{i} - \epsilon\right)\alpha_{i,i}, \quad \text{for } i = 5, 6.$$

The restricted phases of the \mathcal{A}_{gps} algorithm also impose the constraint that the corresponding collections of sets are selected accordingly in order to guarantee that the number of 1-sets at the end of the execution is not larger than that of \mathcal{O}_i , for $i = 6, 5, 4$. The same restriction is indirectly imposed on phase 3 through the semi-local optimization phase. Remember that when attempting to perform an improvement step, the i -set packing algorithm computes the number of 1-sets selected by the semi-local optimization algorithm on the remaining $(\mathcal{U}_{i-1}, \mathcal{C}_{i-1})$ instance. Such a replacement actually takes place only if this number (let that be L_{i-1}) is not larger than the one computed by the semi-local optimization algorithm prior to the replacement. So, it holds that $L_i \geq L_{i-1}$, for $i = 6, \dots, 4$. Also, note that the number of 1-sets computed by the semi-local optimization lower bounds the number of 1-sets in any optimal solution on the corresponding instance. In other words, for each restricted phase i of the algorithm it holds that $\alpha_{i,1} \geq L_i \geq L_3 \geq \alpha_{2,1}$. Summing up, it holds that

$$\alpha_{2,1} \leq \alpha_{i,1}, \quad \text{for } i = 3, \dots, 6.$$

Note that during a semi-local optimization phase, once the 3-sets are chosen, the remaining $(\mathcal{U}_2, \mathcal{C}_2)$ instance is solved optimally. So, we can use variable $\alpha_{2,j}$, for $j = 1, 2$, to denote the ratio of the number of i -sets in the \mathcal{A}_{gps} solution (as well as the optimal solution) over \mathcal{O} . Clearly, it also holds that

$$x_j \leq \alpha_{2,j}, \quad \text{for } j = 1, 2.$$

The semi-local optimization phase naturally has an impact on the number of sets selected by the algorithm. For the sake of clarity denote by b_3, b_2, b_1 the number of 3-, 2-, and 1-sets computed by the semi-local optimization phase divided by $|\mathcal{O}|$. We use the main result of the analysis of Duh and Fürer in [4] expressed in our notation.

Theorem 9 (Duh and Fürer [4]) $b_1 \leq \alpha_{3,1}$ and $b_2 + b_1 \leq \alpha_{3,3} + \alpha_{3,2} + \alpha_{3,1}$.

Let t_3 denote the number of sets selected by the algorithm during the semi-local optimization phase, over $|\mathcal{O}|$. We compute a bound on t_3 .

$$\begin{aligned} t_3 &= (b_3 + b_2 + b_1)|\mathcal{O}| \\ &= \left(\frac{b_1}{3} + \frac{b_2 + b_1}{3} + \frac{3b_3 + 2b_2 + b_1}{3}\right)|\mathcal{O}| \\ &\leq \left(\frac{b_1}{3} + \frac{a_{3,3} + a_{3,2} + a_{3,1}}{3} + \frac{3a_{3,3} + 2a_{3,2} + a_{3,1}}{3}\right)|\mathcal{O}| \\ &= \left(\frac{1}{3}b_1 + \frac{4}{3}a_{3,3} + a_{3,2} + \frac{2}{3}a_{3,1}\right)|\mathcal{O}|, \end{aligned}$$

where, the inequality follows by Theorem 9 and since $(3b_3 + 2b_2 + b_1)|\mathcal{O}| = (3a_{3,3} + 2a_{3,2} + a_{3,1})|\mathcal{O}| = |\mathcal{U}_3|$.

Next, we compute a bound on the total number of sets selected by the algorithm. Recall that b_1 is by definition equal to $\alpha_{2,1}$. We get

$$\begin{aligned} \mathcal{S}_{gps} &\leq \left(\frac{1}{7}|\mathcal{U} \setminus \mathcal{U}_6| + \sum_{i=4}^6 \frac{1}{i}|\mathcal{U}_i \setminus \mathcal{U}_{i-1}| + t_3 \right) / |\mathcal{O}| \\ &= \frac{1}{7} \left(T - \sum_{j=1}^7 j\alpha_{4,j} \right) + \sum_{i=4}^6 \frac{1}{i} \left(\sum_{j=1}^i j\alpha_{i,j} - \sum_{j=1}^{i-1} j\alpha_{i-1,j} \right) + \frac{4}{3}a_{3,3} + a_{3,2} + \frac{2}{3}a_{3,1} + \frac{1}{3}\alpha_{2,1} \\ &= \frac{1}{7}T + \sum_{i=4}^6 \frac{1}{i(i+1)} \sum_{j=1}^i j\alpha_{i,j} + \frac{7f}{12}\alpha_{3,3} + \frac{f}{2}\alpha_{3,2} + \frac{5f}{12}\alpha_{3,1} + \frac{1}{3}\alpha_{2,1}. \end{aligned}$$

The profit obtained by the \mathcal{A}_{gps} algorithm and the optimal algorithm on instance \mathcal{I} , respectively, is expressed through the following inequalities

$$\mathcal{A}_{gps}(\mathcal{I}) = \left(T + 1 - \sum_{i=1}^7 x_i \right) |\mathcal{O}|, \text{ and}$$

$$\text{OPT}(\mathcal{I}) = \left(T + \mathcal{S}_{gps} - \sum_{i=1}^7 x_i \right) |\mathcal{O}|.$$

Finally, recall our initial assumption which states that $\mathcal{A}_{gps}(\mathcal{I}) \leq f \cdot \text{OPT}(\mathcal{I})$.

We are now ready to state our parameterized LP lemma.

Lemma 10 *If there exists an instance \mathcal{I} of the frugal coverage problem for which \mathcal{A}_{gps} computes a solution of benefit $\mathcal{A}_{gps}(\mathcal{I}) \leq f \cdot \text{OPT}(\mathcal{I})$ for some $f \in [0, 1]$, then the linear program $\text{LP}_{gps}(f)$ has a feasible solution.*

$$\begin{aligned} \sum_{j=1}^i \alpha_{i,j} + \sum_{j=i+1}^7 x_j &\leq 1, \quad \text{for } i = 2, \dots, 6 \\ - \left(4 - \frac{7}{4}\right)\alpha_{4,4} - \sum_{j=1}^3 j\alpha_{4,j} + \sum_{j=1}^3 j\alpha_{3,j} &\leq 0 \\ - (i - 2 + i\epsilon)\alpha_{i,i} - \sum_{j=1}^{i-1} j\alpha_{i,j} + \sum_{j=1}^{i-1} j\alpha_{i-1,j} &\leq 0, \quad \text{for } i = 5, 6 \\ - \sum_{j=1}^i j\alpha_{i,j} + \sum_{j=1}^{i-1} j\alpha_{i-1,j} + ix_i &\leq 0, \quad \text{for } i = 3, \dots, 6 \\ - T + \sum_{j=1}^6 j\alpha_{6,j} + 7x_7 &\leq 0 \\ - \alpha_{2,j} + x_j &\leq 0, \quad \text{for } j = 1, 2 \\ - \alpha_{i,1} + \alpha_{2,1} &\leq 0, \quad \text{for } i = 3, \dots, 6 \end{aligned}$$

$$(1 - 8f/7)T - \sum_{i=4}^6 \sum_{j=1}^i \frac{fj}{i(i+1)} \alpha_{i,j} - \frac{7f}{12} \alpha_{3,3} - \frac{f}{2} \alpha_{3,2} - \frac{5f}{12} \alpha_{3,1} - \frac{f}{3} \alpha_{2,1} - \sum_{i=1}^7 (1-f)x_i \leq -1$$

$$\alpha_{i,j} \geq 0, \quad \text{for } i = 2, \dots, 6 \text{ and } j = 1, \dots, i$$

$$x_j \geq 0, \quad \text{for } j = 1, \dots, 7$$

$$T \geq 0$$

The proof of the approximation bound is based on the following lemma.

Lemma 11 *For every $f < 75/86$, $LP_{gps}(f)$ has no feasible solution.*

Proof. The proof follows along similar lines as the one of Lemma 2. Assuming, without loss of generality, that $LP_{gps}(f)$ is a maximization linear program with objective 0, its dual should be a minimization linear program with objective value 0 as well. We compute the dual of $LP_{gps}(f)$ and show that it has a solution with strictly negative objective value for $f < 75/86$. This implies the lemma.

To construct the dual of $LP_{gps}(f)$ we use the twenty variables β_i , for $i = 2, \dots, 6$, corresponding to the first batch of constraints of $LP_{gps}(f)$, γ_i , for $i = 4, \dots, 6$, corresponding to the i -set packing approximation constraints, δ_i , for $i = 3, \dots, 6$, corresponding to the next batch of constraints, η for the next constraint, λ_i , for $i = 1, 2$, for the constraints stemming from the optimality of the solution of the $(\mathcal{U}_2, \mathcal{C}_2)$ instance, μ_i , for $i = 3, \dots, 6$, corresponding to the constraints regarding the restriction of the number of 1-sets and ζ corresponding to the last constraint. The dual of $LP_{gps}(f)$ is presented in Table 3.

$$\begin{array}{ll}
\min & \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 - \zeta \\
\text{s.t.} & \beta_2 + \delta_3 + \mu_3 + \mu_4 + \mu_5 + \mu_6 - \lambda_1 - \frac{f}{3}\zeta \geq 0 \quad \beta_6 - \gamma_6 - \delta_6 + \eta - \mu_6 - \frac{f}{42}\zeta \geq 0 \\
& \beta_2 + 2\delta_3 - \lambda_2 \geq 0 \quad \beta_6 - 2\gamma_6 - 2\delta_6 + 2\eta - \frac{f}{21}\zeta \geq 0 \\
& \beta_3 + \gamma_4 - \delta_3 + \delta_4 - \mu_3 - \frac{5f}{12}\zeta \geq 0 \quad \beta_6 - 3\gamma_6 - 3\delta_6 + 3\eta - \frac{f}{14}\zeta \geq 0 \\
& \beta_3 + 2\gamma_4 - 2\delta_3 + 2\delta_4 - \frac{f}{5}\zeta \geq 0 \quad \beta_6 - 4\gamma_6 - 4\delta_6 + 4\eta - \frac{2f}{7}\zeta \geq 0 \\
& \beta_3 + 3\gamma_4 - 3\delta_3 + 3\delta_4 - \frac{7f}{12}\zeta \geq 0 \quad \beta_6 - 5\gamma_6 - 5\delta_6 + 5\eta - \frac{5f}{42}\zeta \geq 0 \\
& \beta_4 - \gamma_4 + \gamma_5 - \delta_4 + \delta_5 - \mu_4 - \frac{f}{20}\zeta \geq 0 \quad \beta_6 - (4 + 6\epsilon)\gamma_6 - 6\delta_6 + 6\eta - \frac{f}{7}\zeta \geq 0 \\
& \beta_4 - 2\gamma_4 + 2\gamma_5 - 2\delta_4 + 2\delta_5 - \frac{f}{10}\zeta \geq 0 \quad \lambda_1 - (1-f)\zeta \geq 0 \\
& \beta_4 - 3\gamma_4 + 3\gamma_5 - 3\delta_4 + 3\delta_5 - \frac{3f}{20}\zeta \geq 0 \quad \lambda_2 - (1-f)\zeta \geq 0 \\
& \beta_4 - (4 - \frac{7}{4})\gamma_4 + 4\gamma_5 - 4\delta_4 + 4\delta_5 - \frac{f}{5}\zeta \geq 0 \quad \beta_2 + 3\delta_3 - (1-f)\zeta \geq 0 \\
& \beta_5 - \gamma_5 + \gamma_6 - \delta_5 + \delta_6 - \mu_5 - \frac{f}{30}\zeta \geq 0 \quad \beta_2 + \beta_3 + 4\delta_4 - (1-f)\zeta \geq 0 \\
& \beta_5 - 2\gamma_5 + 2\gamma_6 - 2\delta_5 + 2\delta_6 - \frac{f}{15}\zeta \geq 0 \quad \beta_2 + \beta_3 + \beta_4 + 5\delta_5 - (1-f)\zeta \geq 0 \\
& \beta_5 - 3\gamma_5 + 3\gamma_6 - 3\delta_5 + 3\delta_6 - \frac{f}{10}\zeta \geq 0 \quad \beta_2 + \beta_3 + \beta_4 + \beta_5 + 6\delta_6 - (1-f)\zeta \geq 0 \\
& \beta_5 - 4\gamma_5 + 4\gamma_6 - 4\delta_5 + 4\delta_6 - \frac{2f}{15}\zeta \geq 0 \quad \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 + 7\eta - (1-f)\zeta \geq 0 \\
& \beta_5 - (3 + 5\epsilon)\gamma_5 + 5\gamma_6 - 5\delta_5 + 5\delta_6 - \frac{f}{6}\zeta \geq 0 \quad -\eta + (1 - \frac{8f}{7})\zeta \geq 0 \\
& \beta_i \geq 0, \quad \text{for } i = 2, \dots, 6 \\
& \gamma_i \geq 0, \quad \text{for } i = 4, \dots, 6 \\
& \delta_i, \mu_i \geq 0, \quad \text{for } i = 3, \dots, 6 \\
& \lambda_i \geq 0, \quad \text{for } i = 1, 2 \\
& \zeta, \eta \geq 0
\end{array}$$

Table 3: The dual of $LP_{gps}(f)$ in the proof of Lemma 10.

The solution $\beta_2 = 11f/75, \beta_3 = f/3, \beta_4 = 2f/5, \beta_5 = 0, \beta_6 = 4f/15, \gamma_4 = f/12, \gamma_5 = 0, \gamma_6 = f/30, \delta_i = 0$, for $i = 3, \dots, 6, \eta = 1 - 8f/7, \mu_3 = \mu_5 = 0, \mu_4 = 4f/15, \mu_6 = f/15, \lambda_1 = \lambda_2 = 1 -$

f and $\zeta = 1$ satisfies all the constraints, for $\epsilon < 1/15$. Observe that $\beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 - \zeta = \frac{86f}{75} - 1$, which implies that the objective value of the dual program is strictly negative for $f < \frac{75}{86}$. The lemma follows. ■

Theorem 12 *The approximation ratio of \mathcal{A}_{gps} is at least $75/86$.*

Proof. By Lemmas 10 and 11, we have that for any $f < 75/86$ and for any instance \mathcal{I} of the frugal coverage problem, the greedy algorithm with the restricted set packing phases and the semi-local optimization phase computes a solution of benefit $\mathcal{A}_{gps}(\mathcal{I}) > f \cdot \text{OPT}(\mathcal{I})$. Hence, its approximation ratio is at least $75/86$. ■

4.2 The upper bound

We use the construction of Hurkens and Schrijver [9] for the local search heuristic for 5-set packing to prove that our analysis is tight. We describe the instance that yields the upper bound as follows. Let there be $75k$ elements, where k is a sufficiently large integer. These elements are contained in $15k$ disjoint sets of size 5, which we call the vertical sets and denote the optimal solution. The instance has $6k$ additional disjoint horizontal sets of size 5, such that each set overlaps exactly five vertical sets, and two elements from each vertical set are contained in horizontal ones. The construction in [9] guarantees that these horizontal sets can be defined in such a way that they form a locally optimal solution for the local search heuristic. The instance is completed by $5k$ disjoint horizontal sets of size 3, such that each set overlaps exactly three vertical ones and no horizontal set of size 5. See Figure 3 for an example. Note that, for the sake of clarity, the horizontal 5-sets are shown to contain consecutive elements.

We now present the solution returned by \mathcal{A}_{gps} . The execution of the algorithm starts at phase 5, where it selects all horizontal 5-sets. Remember that it follows from the construction that the collection of the horizontal 5-sets is a local optimum for the restricted 5-set packing algorithm of Furer and Yu [7]. The algorithm now applies the semi-local optimization phase, since no 4-set exists in the remaining instance. During this phase \mathcal{A}_{gps} first selects all horizontal 3-sets, and completes the cover using all vertical sets. Summing up, the algorithm uses all available sets in order to cover the $75k$ elements, while the optimal solution uses only vertical sets, obtaining a profit of $75k + 11k = 86k$. This yields an approximation ratio of $75/86$ as desired.

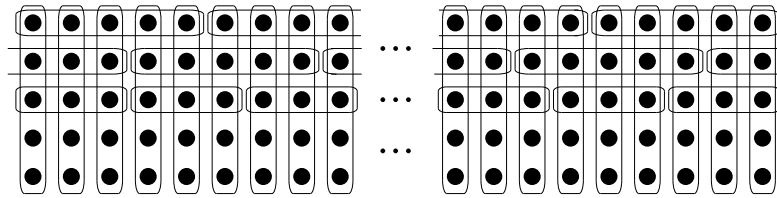


Figure 3: The tight upper bound for \mathcal{A}_{gps} .

5 Open problems

Our focus in the current paper has been on combinatorial greedy-like algorithms for frugal coverage. We have demonstrated how to obtain tight bounds on the approximation ratio of these algorithms using an analysis based on parameterized LPs. The obvious open problem is to design algorithms with better approximation ratio than 0.872. Towards this goal we believe that new algorithmic techniques are required; such techniques could possibly lead to improvements for other covering problems as well.

References

- [1] S. Athanassopoulos, I. Caragiannis, and C. Kaklamanis. Analysis of approximation algorithms for k -set cover using factor-revealing linear programs. *Theory of Computing Systems*, 45(3), pp. 555-576, 2009.
- [2] S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and M. Kyropoulou. An improved approximation bound for spanning star forest and color saving. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, LNCS 5734, Springer, pp. 90-101, 2009.
- [3] I. Caragiannis. Wavelength management in WDM rings to maximize the number of connections. *SIAM Journal on Discrete Mathematics*, 23(2), pp. 959-978, 2009.
- [4] R. Duh and M. Fürer. Approximation of k -set cover by semi-local optimization. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 256-264, 1997.
- [5] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4), pp. 634-652, 1998.
- [6] U. Feige and S. Jozeph. Oblivious algorithms for the maximum directed cut problem. *arXiv: 1010.0406*, 2010.
- [7] M. Fürer and H. Yu. Packing-Based Approximation Algorithm for the k -Set Cover Problem. In *Proceedings of the the 22nd International Symposium on Algorithms and Computation (ISAAC '11)*, 2011, to appear.
- [8] C.-C. Huang and Z. Svitkina. Donation center location problem. In *Proceedings of the 29th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pp. 227-238, 2009.
- [9] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1), pp. 68-72, 1989.
- [10] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6), pp. 795-824, 2003.
- [11] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9, pp. 256-278, 1974.
- [12] A. Levin. Approximating the unweighted k -set cover problem: greedy meets local search. *SIAM Journal on Discrete Mathematics*, 23(1), pp. 251-264, 2008.
- [13] A. Levin and U. Yovel. Uniform unweighted set cover: the power of non-oblivious local search. *Theoretical Computer Science*, 412(12-14), pp. 1033-1053, 2011.
- [14] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 475-484, 1997.