The Lazy Matroid Problem

Laurent Gourvès, Jérôme Monnot, Aris Pagourtzis

The Lazy Matroid Problem*

Laurent Gourvès[1,2], Jérôme Monnot[1,2], and Aris T. Pagourtzis[3]

1. CNRS, UMR 7243 (LAMSADE)
2. PSL Université Paris Dauphine
Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16, France
3. National Technical University of Athens (NTUA)
School of Electrical and Computer Engineering
Heroon Politechniou 9, GR-15780 Zographou, Greece
laurent.gourves@dauphine.fr, jerome.monnot@dauphine.fr, pagour@cs.ntua.gr

Abstract. This article introduces the LAZY MATROID PROBLEM, which captures the goal of saving time or money in certain task selection scenarios. We are given a budget B and a matroid $\mathcal{M}$ with weights on its elements. The problem consists in finding an independent set F of minimum weight. In addition, F is feasible if its augmentation with any new element x implies that either $F + x$ exceeds B or $F + x$ is dependent. Our first result is a polynomial time approximation scheme for this **NP**-hard problem which generalizes a recently studied version of the LAZY BUREAUCRAT PROBLEM. We next study the approximability of a more general setting called LAZY STAFF MATROID. In this generalization, every element of $\mathcal{M}$ has a multidimensional weight. We show that approximating this generalization is much harder than for the LAZY MATROID PROBLEM since it includes the INDEPENDENT DOMINATING SET PROBLEM.

Key words: approximation algorithms, matroids, independent dominating set

1 Introduction

Imagine that the Minister of Public Works has to select some projects to fund, among a pool of proposed ones. She has a certain budget that she can spend on these projects and she wants to select projects in such a way that as much money as possible are saved (remain unused), yet not enough for any left-out project. This is in fact a 'reincarnation' of the LAZY BUREAUCRAT PROBLEM [1–5] in which a lazy worker wants to select a set of tasks of minimum total duration in such a way that his remaining working time does not suffice to add any task.

* Aris Pagourtzis was supported for this research by the project ALGONOW, co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALES, Investing in knowledge society through the European Social Fund.

Assume further that the Minister has to deal with additional constraints, e.g. if the country is divided into regions and there is a maximum number of projects that should be allocated per region. Such constraints can often be described by a matroid on the set of tasks: for example, the above case can be described by a partition matroid. To address such scenarios, we define a generalization of the Lazy Bureaucrat Problem, which we call Lazy Matroid: given a weighted matroid of tasks $(S, \mathcal{F})$ and a budget B, we want to select a set of tasks $S' \subseteq S$ such that adding every left-out task to S' would violate either the budget B or the matroid constraint (or both); the goal is to minimize the weight of S'.

Another situation that can be well described by the Lazy Matroid problem concerns a network design problem: one wants to connect several parts of a network by, say, optical fibers, without exceeding a given cost budget, and respecting two constraints: it is not allowed to create cycles (this is considered unnecessary spending) and it is not acceptable to avoid establishing a connection between unconnected components, if the remaining budget suffices. It is reasonable to assume that the network manager would like to spend as little as possible, without violating the constraints. This is an instance of the Lazy Matroid Problem on graphic matroids.

Our contribution In this work we formally define and study the Lazy Matroid problem, which is **NP**-hard, since so is the Lazy Bureaucrat Problem; the latter is a special case of the former as pointed out above. Our first result is a *PTAS* for Lazy Matroid. The proposed algorithm involves careful employment of two well known greedy algorithms for weighted matroids, in conjunction with appropriately designed matroid contraction and restriction operations.

We next consider a more general setting, in which each task has to be carried out by several workers, who collectively wish to minimize their total work load; we call this variant Lazy Staff Matroid. In the Public Works scenario, this would correspond to projects associated with multiple weights, each representing an estimation of the project's negative impact in some domain of increased importance: environment, cultural heritage, unemployment, to name a few; then, one might want to bound the total impact of selected projects in each of the considered domains.

In contrast to the one-worker case, we show that Lazy Staff Matroid is highly inapproximable. We do this by reduction from the independent dominating set problem (ISDS in short). Along the way, we obtain some new (to the best of our knowledge) inapproximability results for ISDS on regular graphs. We finally present a $2m$-approximation algorithm for Lazy Staff Matroid on free matroids. Some proofs are omitted due to space limitation.

Related work To the best of our knowledge, this is the first study on the matroidal version of the Lazy Bureaucrat scheduling problem; the latter was defined by Arkin, Bender, Mitchell and Skiena [6, 1] under various optimization objectives. In fact, Lazy Matroid is a generalization of Lazy Bureaucrat with common arrivals and deadlines, which was shown to admit an *FPTAS* in [2]; note that in the common arrivals case the two most studied objectives, namely `makespan` and `time-spent` coincide. The (weak) **NP**-hardness of this case was shown by Gai and Zhang [7, 3].

Earlier results on Lazy Bureaucrat include approximations for the common deadline case: first a tight 2-approximation algorithm working under both objectives was given by Esfahbod, Ghodsi and Sharifi [8] and later two *PTAS*'s, one for each objective, were presented in [7, 3].

Note that the Lazy Bureaucrat with common arrivals and deadlines is a knapsack-like problem with an inverted objective function since one tries to *minimize* the total value of the solution. Camerini and Vercellis have studied a matroidal version of the knapsack with its classical objective function of *maximizing* the total value of the solution [9].

Coming to the Lazy Staff Matroid problem, we show in Section 5 that it includes well-known problems as special cases, most notably the independent dominating set problem (ISDS). ISDS is not approximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ on graphs of n vertices [10] (unless $\mathbf{P} = \mathbf{NP}$). In addition, it is **NP**-hard to approximate ISDS in graphs of degree at most 3 within a factor $\frac{681}{680}$ [11]. Regarding regular graphs no approximability hardness results for ISDS can be found in the literature, up to our best knowledge.

2 Basic notions on matroids

This section comprises basic notions on matroids, see [12, 13] for more details. We use the shorthand notation $X + x := X \cup \{x\}$ and $X - y := X \setminus \{y\}$. A matroid $\mathcal{M} = (X, \mathcal{F})$ is a finite set of elements X and a collection $\mathcal{F}$ of subsets of X satisfying the following properties: (i) $\emptyset \in \mathcal{F}$; (ii) if $F_2 \subseteq F_1$ and $F_1 \in \mathcal{F}$ then $F_2 \in \mathcal{F}$; (iii) for every $F_1, F_2 \in \mathcal{F}$ where $|F_1| < |F_2|$, $\exists\ e \in F_2 \backslash F_1$ such that $F_1 + e \in \mathcal{F}$.

The elements of $\mathcal{F}$ and $2^X \setminus \mathcal{F}$ are called *independent* sets and *dependent* sets, respectively. The *bases* of a matroid are its inclusion-wise maximal independent sets. All bases of a matroid $\mathcal{M}$ have the same cardinality $r(\mathcal{M})$, defined as the *rank* of $\mathcal{M}$.

In the presence of a weight function $w : X \to \mathbb{R}$, we use the shorthand notation $w(X') = \sum_{x \in X'} w(x)$ for all $X' \subseteq X$. A matroid $(X, \mathcal{F})$ where each element e has a weight $w(e)$ is a *weighted matroid*; it is denoted by $(X, \mathcal{F}, w)$.

Given $(X, \mathcal{F}, w)$, a classical optimization problem consists in computing a base of minimum weight. This problem is solved by Min-Greedy (see Algorithm 1). Computing a base of *maximum weight* can be done with a similar algorithm called Max-Greedy (the elements of $\mathcal{M}$ are scanned by non-increasing weight); the output of Max-Greedy($\mathcal{M}$) is denoted by $\max -Gr(\mathcal{M})$.

The time complexity of matroid algorithms depends on the difficulty of testing if a set F belongs to $\mathcal{F}$. We deliberately neglect this test when the time complexity of an algorithm is provided. Thus, Algorithm 1 runs in polynomial time.

Given a matroid $\mathcal{M} = (X, \mathcal{F})$ and $Y \subseteq X$, the *restriction* of $\mathcal{M}$ to Y, denoted by $\mathcal{M}|Y$, is the structure $(Y, \mathcal{F}')$ where $\mathcal{F}' = \{Z \subseteq Y : Z \in \mathcal{F}\}$. If $Y \in \mathcal{F}$, the *contraction* of $\mathcal{M}$ by Y, denoted by $\mathcal{M}/Y$, is the structure $(X \setminus Y, \mathcal{F}')$ where $\mathcal{F}' = \{F \subseteq X \setminus Y : F \cup Y \in \mathcal{F}\}$. It is well known that both $\mathcal{M}|Y$ and $\mathcal{M}/Y$ are matroids.

Next Theorem is satisfied by any matroid $\mathcal{M}$.

Theorem 1. *[14] Let B and B' be bases and let $x \in B - B'$. Then there exists $y \in B' - B$ such that both $B - x + y$ and $B' - y + x$ are bases of $\mathcal{M}$.*

Matroids are known to model several structures in combinatorial optimization. For instance, the *free matroid* is defined on a set X, each subset $F \subseteq X$ is independent and the unique base is X. A second example is the *graphic matroid* which is defined on the set of edges of a graph G, the independent sets are the forests of G (subsets of edges without cycles). A base of the graphic matroid is a spanning tree if the graph G is connected. A third example is the *partition matroid*; this matroid is defined on a set X partitioned into k disjoint sets $X_1, \ldots, X_k$ for $k \geq 1$. Given k integers $b_i \geq 0$ ($i = 1, ..., k$), the independent sets are all the sets $F \subseteq X$ satisfying $|F \cap X_i| \leq b_i$ for all $i = 1, \ldots, k$.

Algorithm 1: MIN-GREEDY

Data: a weighted matroid $\mathcal{M} = (X, \mathcal{F}, w)$

1 Rename $X = \{x_1, \cdots, x_n\}$ such that $w(x_i) \leq w(x_{i+1})$, $i \leq n - 1$

2 $Gr(\mathcal{M}) \leftarrow \emptyset$

3 **for** $i = 1$ *to* n **do**

4   **if** $Gr(\mathcal{M}) \cup \{x_i\} \in \mathcal{F}$ **then**

5         $Gr(\mathcal{M}) \leftarrow Gr(\mathcal{M}) \cup \{x_i\}$

6 **return** $Gr(\mathcal{M})$

Note that MIN-GREEDY and MAX-GREEDY can also be used to complete an independent set $F \in \mathcal{F}$ into a base. Instead of starting with the empty set as in step 2 of the algorithms, we begin with F. Thus, the completion of F with MIN-GREEDY and MAX-GREEDY provides a base of minimum and maximum weight, respectively, within the set of bases which contain F.

3 Problem Definition & Properties

LAZY MATROID PROBLEM

Input: a weighted matroid $\mathcal{M} = (X, \mathcal{F}, w)$, where w is a positive weight function $w : X \to \mathbb{R}^+$ and a positive bound B.

Output: $F \in \mathcal{F}$ with $w(F) \leq B$ and s.t. $\forall x \in X \setminus F, F + x \in \mathcal{F} \Rightarrow w(F + x) > B$.

Objective: minimize $w(F)$.

Note that a feasible solution to Lazy Matroid Problem must satisfy a constraint of maximality which counterbalances the fact that the weight of a solution must be minimized. This constraint refers to the *busy requirement* of the Lazy Bureaucrat Problem [1].

In what follows, all solution sets will be assumed to be sorted in non decreasing order of weight, unless otherwise stated. For $t \leq n$, $X_t = \{x_1, \dots, x_t\}$ is the restriction of X to the t smallest elements and $\mathcal{M}_t$ is the restriction of $\mathcal{M}$ to X_t. It is well known that $\mathcal{M}_t$ remains a matroid.

Let $OPT(\mathcal{M}, B) = \{x_{\pi(1)}, \dots, x_{\pi(p)}\}$ be an optimal solution to the Lazy Matroid Problem on instance $(\mathcal{M}, B)$. We will omit $(\mathcal{M}, B)$ when the context is clear ; $p = |OPT|$. For $t \leq p$, $OPT_t = \{x_{\pi(1)}, \dots, x_{\pi(t)}\}$ is the restriction of OPT to the t smallest elements.

$Gr(\mathcal{M})$ is the solution returned by the greedy algorithm Min-Greedy with weighted matroid $\mathcal{M}$, see Algorithm 1. It is well known that $Gr(\mathcal{M})$ is a base of $\mathcal{M}$ and has a minimum weight among all bases of $\mathcal{M}$. Actually more generally, if $Gr_t(\mathcal{M})$ denotes the restriction of $Gr(\mathcal{M})$ to the t first elements taken by Min-Greedy, then $Gr_t(\mathcal{M})$ has a minimum weight among all independent sets of $\mathcal{M}$ with size exactly t. Finally $\max -Gr(\mathcal{M})$ is a base of maximum weight of $\mathcal{M}$ and it is returned by Max-Greedy algorithm.

Lazy Greedy is an adaptation of Greedy for the Lazy Matroid Problem and it is described in Algorithm 2.

Algorithm 2: Lazy Greedy

Data: a weighted matroid $\mathcal{M} = (X, \mathcal{F}, w)$ and a bound B

```
Rename X = {x_1, ..., x_n} such that w(x_i) ≤ w(x_{i+1}), i ≤ n − 1
LazyGr(M, B) ← ∅
for i = 1 to n do
    if LazyGr(M, B) ∪ {x_i} ∈ F and w(LazyGr(M, B)) ≤ B then
        LazyGr(M, B) ← LazyGr(M, B) ∪ {x_i}
return LazyGr(M, B)
```

On free matroids, Lazy Greedy coincides with the shortest job first scheduling policy introduced in [8] for the common deadline case of the Lazy Bureaucrat Problem and provides, in the worst case, a 2-approximation [8]. As proved in [2], a slight modification of this greedy algorithm gives a 4/3-approximation for Lazy Bureaucrat Problem in linear time.

Note that $LazyGr(\mathcal{M}, B)$ is a feasible solution to the Lazy Matroid Problem, but it does not guarantee any constant approximation ratio.

Now, we give some properties on Lazy Greedy which will be useful later. We will suppose that $LazyGr(\mathcal{M}, B) = \{x_{f(1)}, \dots, x_{f(s)}\}$ is the solution returned by Lazy Greedy; $s = |LazyGr(\mathcal{M}, B)|$. Note that we have $LazyGr(\mathcal{M}, B) = Gr_s(\mathcal{M})$ under the previous notations.

Lemma 1. *Let $k \geq 1$ be an integer. If $w(OPT) \geq B\frac{k}{k+1}$ then* Lazy Greedy *is a $\frac{k+1}{k}$-approximation.*

Lemma 2. *For any instance $(\mathcal{M}, B)$, we have $s \geq p$.*

Actually, the case $s = p$ is polynomially solvable by Lazy Greedy. Henceforth, we focus on the case $s > p$.

4 A *PTAS*

Let us give an overview of the PTAS. Given k in input, the algorithm consists in testing every possible subset of *at most* k elements. Each of these sets that satisfies the feasibility constraint of the lazy matroid problem is stored in a set denoted by Sol. If the optimum uses at most k elements then it must belong to Sol. Otherwise, one tries to guess $A^* = \{x_{g^{A^*}(1)}, \ldots, x_{g^{A^*}(k)}\}$, the k elements of OPT with largest weight. Then $\mathcal{M}$ is contracted by A^* and restricted to the elements of X whose weight does not exceed the weight of the lightest element of A^*. This matroid is denoted by $\mathcal{M}^{A^*}$. max-greedy is run on $\mathcal{M}^{A^*}$ in order to get a set $\{x_{h^{A^*}(1)}, \ldots, x_{h^{A^*}(s^{A^*})}\}$ of s^{A^*} elements. Then $s^{A^*} + 1$ sets F_0, F_1, $\ldots F_{s^{A^*}}$ are constructed as follows. For $t \in \{0, 1, \ldots, s^{A^*}\}$, min-greedy is run to complete $\{x_{h^{A^*}(s^{A^*}-t+1)}, \ldots, x_{h^{A^*}(s^{A^*})}\}$ into a base F_t of $\mathcal{M}^{A^*}$. Every set $F_t + A^*$ that satisfies the feasibility constraint of the lazy matroid problem on $(\mathcal{M}, B)$ is added to Sol. Finally, the solution of minimum weight stored in Sol is returned. The algorithm, formally described in Algorithm 3, is shown to be $\frac{k+1}{k}$-approximate.
Note that Sol $\neq \emptyset$ because PTAS-Lazy contains at least the Lazy Greedy solution on initial instance $(\mathcal{M}, B)$. Indeed, when A' denotes the set of the k heaviest elements of $LazyGr(\mathcal{M}, B)$, we have $A' + U_0^{A'} + Gr(\mathcal{M}^{A',0}) = LazyGr(\mathcal{M}, B)$ for iteration $t = 0$.

Let us fix an integer $k \geq 1$ and let us prove that PTAS-Lazy (Algorithm 3) with input k is a $\frac{k+1}{k}$-approximation. Let APX be the solution returned by PTAS-Lazy on input $(\mathcal{M}, B, k)$. Let $OPT = \{x_{\pi(1)}, \ldots, x_{\pi(p)}\}$ be an optimal solution satisfying $\pi(1) < \ldots < \pi(p)$ and $|OPT| = p$. If $|OPT| \leq k$ then $OPT \in$ Sol and the algorithm is 1-approximate. Suppose from now on that $p = |OPT| > k$. Let A^* be the k heaviest elements of OPT, i.e. $A^* = \{x_{\pi(p-k+1)}, \ldots, x_{\pi(p)}\}$. Following the notations of Algorithm 3, we can also define A^* as $\{x_{g^{A^*}(1)}, \ldots, x_{g^{A^*}(k)}\}$. Let $\mathcal{M}^{A^*}$ denote $(X_{g^{A^*}(1)-1}, \mathcal{F}^{A^*}, w)$ where $\mathcal{F}^{A^*}$ is the restriction of $\mathcal{F}$ to the subsets of $X_{g^{A^*}(1)-1} = \{x_1, \ldots, x_{g^{A^*}(1)-1}\}$.

Lemma 3. *If $w(OPT) < B\frac{k}{k+1}$ then $OPT - A^*$ is a base of $\mathcal{M}^{A^*}$.*

For $t = 0, \ldots, s^{A^*}$, let F_t be a base of $\mathcal{M}^{A^*}$ defined as $F_t = U_t^{A^*} \cup Gr(\mathcal{M}^{A^*,t})$. Following Algorithm 3, $U_t^{A^*}$ consists of the t heaviest elements of $\max -Gr(\mathcal{M}^{A^*})$ whereas $Gr(\mathcal{M}^{A^*,t})$ is obtained by running min-greedy on $\mathcal{M}^{A^*,t}$.

Note that $F_{s^{A^*}} = \max -Gr(\mathcal{M}^{A^*})$ and $F_0 = Gr(\mathcal{M}^{A^*})$. Moreover, by a property of Min-Greedy, we know that for any t, F_t is a base of $\mathcal{M}^{A^*}$ with minimum weight among the bases of $\mathcal{M}^{A^*}$ containing $U_t^{A^*}$ (note that $U_0^{A^*} = \emptyset$).

Algorithm 3: PTAS-Lazy

Data: a weighted matroid $\mathcal{M} = (X,\ \mathcal{F}, w)$, a bound B and an integer $k \geq 1$

Rename $X = \{x_1, \cdots, x_n\}$ such that $w(x_i) \leq w(x_{i+1})$, $i \leq n-1$

for *all* $A \subseteq X$ *of size at most* k **do**

  if A *is a feasible solution to the* LAZY MATROID *problem for instance* $(\mathcal{M}, B)$ **then**

    Sol $\leftarrow$ Sol $+ A$

for *all* $A = \{x_{g^A(1)}, \ldots, x_{g^A(k)}\} \subseteq X$ *and* $A \notin$ Sol *with* $|A| = k$, $w(A) < B$ *and* $g^A(1) < \cdots < g^A(k)$ **do**

  Let $\mathcal{M}^A$ be the matroid restricted to $X_{g^A(1)-1}$ and contracted to A

  Compute $\max -Gr(\mathcal{M}^A) = \{x_{h^A(1)}, \ldots, x_{h^A(s^A)}\}$, a maximum weight base of $\mathcal{M}^A$ where $h^A(1) < \ldots < h^A(s^A)$

  for $t = 0$ *to* $s^A = |\max -Gr(\mathcal{M}^A)|$ **do**

    Let U_t^A be the t heaviest elements of $\max -Gr(\mathcal{M}^A)$

    For $t \geq 1$, let $\mathcal{M}^{A,t} = \mathcal{M}_{h^A(s^A-t)}|U_t^A$ be $\mathcal{M}^A$ restricted to $X_{h^A(s^A-t)}$ and contracted to U_t^A; for $t = 0$, let $\mathcal{M}^{A,0} = \mathcal{M}^A$ and $U_0^A = \emptyset$

    if $A + U_t^A + Gr(\mathcal{M}^{A,t})$ *is a feasible solution to the* LAZY MATROID *problem on instance* $(\mathcal{M}, B)$ **then**

      Sol $\leftarrow$ Sol $+ \left(A + U_t^A + Gr(\mathcal{M}^{A,t})\right)$

return the best solution within Sol

Lemma 4. *If* $w(OPT) < B\frac{k}{k+1}$ *then either* $F_0 + A^*$ *is an optimal solution to the* LAZY MATROID *problem on* $(\mathcal{M}, B)$*, or there exists* $t > 1$ *such that* $w(F_t) \geq w(OPT - A^*)$ *and* $w(F_{t-1}) < w(OPT - A^*)$*.*

Since both F_t and F_{t-1} are bases of $\mathcal{M}^{A^*}$, we can use Theorem 1 with $x_{h^{A^*}(s^{A^*}-t+1)} \in F_t \setminus F_{t-1}$ to state that there must be $a \in F_{t-1} \setminus F_t$ such that $U := F_{t-1} + x_{h^{A^*}(s^{A^*}-t+1)} - a$ is also a base of $\mathcal{M}^{A^*}$.

Lemma 5. *If* $w(OPT) < B\frac{k}{k+1}$ *then* $F_t + A^*$ *is a* $\frac{k+1}{k}$*-approximate solution to the* LAZY MATROID *problem on* $(\mathcal{M}, B)$*.*

Proof. Both F_t and U contain $U_t^{A^*}$ and they are both bases of $\mathcal{M}^{A^*}$. Since F_t is a base of $\mathcal{M}^{A^*}$ with minimum weight among the bases of $\mathcal{M}^{A^*}$ containing $U_t^{A^*}$, we get that $w(F_t) \leq w(U)$. Since neither F_t nor U contains A^*, we deduce that

$$w(F_t + A^*) \leq w(U + A^*) \tag{1}$$

We can also see that $w(U) \leq w(F_{t-1}) + w(x_{h^{A^*}(s^{A^*}-t+1)}) < w(OPT - A^*) + \frac{1}{k}w(OPT)$. The first inequality follows from the definition of U. The second inequality is due to $w(F_{t-1}) < w(OPT - A^*)$ (Lemma 4) and the fact that $\forall x \in X_{g^{A^*}(1)-1}$, $kw(x) \leq w(A^*) \leq w(OPT)$. Hence,

$$w(U + A^*) \leq w(OPT) + \frac{1}{k}w(OPT) \tag{2}$$

Using Inequalities (1) and (2), we obtain $w(F_t) \leq \frac{k+1}{k} w(OPT) < B$ because $w(OPT) < B\frac{k}{k+1}$ by hypothesis.

It remains to show that $F_t + A^*$ is a feasible solution to the LAZY MATROID problem. By contradiction, suppose there exists $a \in X \setminus X_{g^{A^*}(1)-1}$ (because F_t is a base of $\mathcal{M}^{A^*}$) such that $F_t + A^* + a \in \mathcal{F}$ and $w(F_t + A^*) + w(a) \leq B$. Note that $|F_t + A^* + a| > |OPT|$ because $|F_t| = |OPT - A^*|$) and $\forall x \in F_t + a$, $w(a) \geq w(x)$. Therefore there exists $b \in (F_t + A^* + a) - OPT$ such that $OPT + b \in \mathcal{F}$ and $w(OPT + b) \leq w(F_t) + w(a) \leq B$, contradicting the feasibility of OPT. □

Theorem 2. PTAS-LAZY *with input k is a polynomial $\frac{k+1}{k}$-approximation for* LAZY MATROID PROBLEM *on $(\mathcal{M}, B)$ where $\mathcal{M} = (X, \mathcal{F}, w)$. The time complexity is $O(|X|^{k+2})$.*

Proof. Let APX be the solution returned by PTAS-LAZY with input k and suppose $|OPT| > k$ (otherwise $w(APX) = w(OPT)$). By construction, $w(APX) \leq B$ because it contains at least one solution (the one returned by LAZY GREEDY). If $w(OPT) \geq B\frac{k}{k+1}$ then $w(APX) \leq \frac{k+1}{k} w(OPT)$. If $w(OPT) < B\frac{k}{k+1}$ then $w(APX) \leq w(F_t + A^*) \leq \frac{k+1}{k} w(OPT)$ by using both $F_t + A^* \in \text{Sol}$ and Lemma 5. In any case, we get the expected result. □

5 The lazy staff matroid problem

LAZY STAFF MATROID PROBLEM
Input: an m-weighted matroid $\mathcal{M} = (X, \mathcal{F}, w)$, where $w : X \to \mathbb{R}_+^m$ is a positive weight function on m dimensions ($w_i(x)$ denotes the i-th component of $w(x)$) and a positive bound B.
Output: $F \in \mathcal{F}$ with $w_i(F) \leq B$ for every $i \in \{1, \ldots, m\}$ and s.t. $\forall x \in X \setminus F, F + x \in \mathcal{F} \Rightarrow w_i(F + x) > B$ for some $i \in \{1, \ldots, m\}$.
Objective: minimize $\sum_{i=1}^{m} w_i(F)$.

For example, dealing with free matroids, a staff is composed of m lazy bureaucrats who have to execute some given jobs. A job is a vector of m non negative integers. Each coordinate k of a job corresponds to the time that worker k would spend for doing his part. In a feasible solution, i.e. a subset of jobs, the constraint of maximality imposes that every additional job would exceed the working time of *at least* one worker.

The LAZY STAFF MATROID PROBLEM is a generalization of the LAZY MATROID PROBLEM; the latter corresponds to the case $m = 1$. The LAZY STAFF MATROID PROBLEM is much harder than the LAZY MATROID PROBLEM.

For instance, it is not difficult to see that the restriction of this problem to binary inputs (i.e., $B, w_i(x) \in \{0, 1\}$) already contains the minimum maximal matching. Given a graph $G = (V, E)$ with n vertices and m edges, instance of MINIMUM MAXIMAL MATCHING, we consider the free matroid on E, $\mathcal{M} = (E, 2^E)$ and we set $B = 1$. There are n mappings w_v for $v \in V$ described by: $w_v(e) = 1$ if v is incident to e in G and $w_v(e) = 0$ otherwise. Since, any Lazy Staff solution corresponds to a maximal matching in G and vice versa, the result follows.

We will prove that the LAZY STAFF MATROID PROBLEM contains the independent dominating set problem (ISDS) in regular graphs. This latter problem is also known as *minimum maximal independent set*. Given a graph $G = (V, E)$, we want to find $S \subset V$ which is *independent* (no two vertices in S are joined by an edge) and *dominating* (every vertex of $V \setminus S$ is adjacent to some vertex of S) of minimum size. ISDS is one of the hardest, well-known, **NP**-hard graph problems. In [10], it is shown that this problem is not approximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ on graphs of n vertices (assuming $\mathbf{P} \neq \mathbf{NP}$). In addition, it is **NP**-hard to approximate ISDS in graphs of degree at most 3 within a factor $\frac{681}{680}$ while a 2-approximation algorithm exists [11]. Up to our best efforts, we were not able to find in the literature any complexity results dealing with regular graphs, but some results can be deduced from existing ones.

Lemma 6. ISDS *is* **APX**-*complete in cubic graphs and it is not constant approximable in regular graphs, unless* $\mathbf{P} = \mathbf{NP}$.

Proof. For the first part of the lemma, we prove that the reduction given in [15, Theorem 13] for the **NP**-completeness of ISDS is actually an L-reduction [16].

First, we start from the dominating set problem (DS) which is known to be **APX**-complete in cubic graphs [17]. Given a cubic graph $G = (V, E)$ with n vertices and m edges, instance of DS, we obtain a cubic graph $H = (V', E')$ by replacing each edge $e = [u, v] \in E$ by a gadget $H(e)$. This transformation is illustrated in Figure 1 and we are going to show that it is an L-reduction.

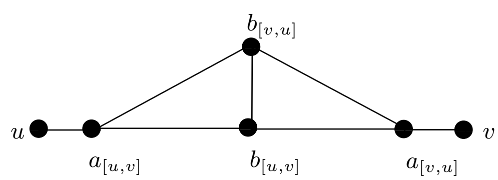

Fig. 1. Local replacement of $[u, v]$ by $H([u, v])$ where four new vertices are added.

Let $\mathcal{D}^*$ be an optimal dominating set of G and let $ds(G)$ denote its size. One can build an independent dominating set $\mathcal{ID}$ of H based on $\mathcal{D}^*$ as follows. Begin with $\mathcal{ID} = \emptyset$. If $v \in \mathcal{D}^*$ then add v to $\mathcal{ID}$. For every edge $[u, v]$ do: if $\mathcal{D}^* \cap \{u, v\} = \emptyset$ then add $b_{[u,v]}$ to $\mathcal{ID}$; if $\mathcal{D}^* \cap \{u, v\} = \{u\}$ then add $a_{[v,u]}$ to $\mathcal{ID}$; if $\mathcal{D}^* \cap \{u, v\} = \{u, v\}$ then add $b_{[u,v]}$ to $\mathcal{ID}$. Thus, an independent dominating set $\mathcal{ID}$ for H is obtained and its size is $ds(G) + m$. Let $isds(H)$ denote the size of an optimal independent dominating set of H. We have:

$$isds(H) \leq ds(G) + m \tag{3}$$

Since G is cubic, we know that $m = 3n/2$ and $ds(G) \geq \frac{n}{4}$ (a node can cover four nodes: itself and its three neighbors). From inequality (3), we get that

$$isds(H) \leq ds(G) + \frac{3n}{2} \leq 7ds(G)$$

From any independent dominating set $\mathcal{ID}$ of H with value $apx(H)$, we can polynomially obtain a dominating set $\mathcal{D}$ of G with value $apx(G)$ satisfying

$$apx(G) \leq apx(H) - m \tag{4}$$

Inequality (4) is obtained as follows. For any edge $[u,v] \in E$, we first observe that we can always suppose that $|\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\}| = 1$. Indeed $|\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\}| \neq 0$ otherwise $b_{[u,v]}$ and $b_{[v,u]}$ are not dominated; $|\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\}| < 3$ otherwise $\mathcal{ID}$ is not independent. It can be $|\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\}| = 2$ only when $\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\} = \{a_{[u,v]}, a_{[v,u]}\}$ and one can modify $\mathcal{ID}$ in order to reduce its size. Let $E'' := \{[u,v] \in E : \mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\} = \{a_{[u,v]}, a_{[v,u]}\}\}$. Proceed as follows until $E'' = \emptyset$:

- take a vertex $s \in V$ endpoint of at least one edge of E'';
- denote by E_s the edges $[s,t] \in E \backslash E''$ such that $\mathcal{ID} \cap \{a_{[s,t]}, a_{[t,s]}, b_{[s,t]}, b_{[t,s]}\} = \{a_{[s,t]}\}$;
- $\mathcal{ID} \leftarrow \mathcal{ID} + s$;
- for every $[s,t] \in E''$, do $\mathcal{ID} \leftarrow \mathcal{ID} - a_{[s,t]}$;
- for every $[s,t] \in E_s$, do $\mathcal{ID} \leftarrow \mathcal{ID} - a_{[s,t]} + b_{[s,t]}$;
- update E'' by deleting all edges incident to s.

The modification is such that $\mathcal{ID}$ remains an independent dominating set with no greater size and $|\mathcal{ID} \cap \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]}\}| = 1$ for any edge $[u,v] \in E$. Moreover $\mathcal{D} = \mathcal{ID} \setminus \{a_{[u,v]}, a_{[v,u]}, b_{[u,v]}, b_{[v,u]} \ : \ [u,v] \in E\}$ is a dominating set of G.

Hence, using inequalities (3) and (4) we deduce that $apx(G) - ds(G) \leq apx(H) - isds(H)$. In conclusion, the reduction is an L-reduction from DS in cubic graphs to ISDS in cubic graphs. Since the former is **APX**-complete [17], we obtain that the latter is (also) **APX**-complete. This implies that no PTAS for ISDS in cubic graphs exists unless $\mathbf{P} = \mathbf{NP}$.

For the second part of the lemma, we use the self improvement of ISDS based on the graph composition as it is done for the independent set problem, see for instance Theorem 6.12, page 146, of [18].

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the *composition* $G_1[G_2]$ is the graph that has vertex set $V_1 \times V_2$ and edge set $\{[(u_1,u_2),(v_1,v_2)]$: either $[u_1,v_1] \in E_1$ or $u_1 = v_1$ and $[u_2,v_2] \in E_2\}$.

Given a *regular* graph $G = (V,E)$ on n vertices and degree $\Delta(G)$, its composition with itself, that is $G[G]$, is denoted by $G'' = (V'', E'')$

It is not difficult to see that G'' is also a regular graph of degree $\Delta(G'') = (n+1)\Delta(G)$ and we have:

$$isds(G'') = isds^2(G) \tag{5}$$

Moreover, from any independent dominating set $IDS(G'')$ of G'' with value $apx(G'')$, we can polynomially find an independent dominating set $IDS(G)$ of G of value $apx(G)$ such that:

$$apx^2(G) \leq apx(G'') \tag{6}$$

Let $D_1 = \{u \in V : (u,v) \in IDS(G'')$ for some $v \in V\}$ and for $u \in D_1$, $D_2^u = \{v \in V : (u,v) \in IDS(G'')\}$. It is easy to check that D_1 and D_2^u for $u \in D_1$ are independent dominating sets of G. Thus, if $IDS(G)$ is the set of smallest cardinality in $\{D_1\} \cup \{D_2^u : u \in D_1\}$ then $apx(G'') = \sum_{u \in D_1} |D_2^u| \geq |IDS(G)|^2 = apx^2(G)$.

In conclusion, any constant approximation of ISDS allows us to obtain a polynomial-time approximation scheme which is a contradiction with the first claim. □

Theorem 3. *Unless* $\mathbf{P} = \mathbf{NP}$*, the* LAZY STAFF MATROID PROBLEM *is not constant approximable even for the free matroid and binary weights (i.e.,* $B, w_i(x) \in \{0,1\}$*).*

Proof. We propose an approximation-preserving reduction from the independent dominating set problem in regular graphs. Let $G = (V, E)$ be a regular graph of degree $\Delta(G)$, with n vertices and m edges, instance of ISDS. Let $\mathcal{M} = (V, 2^V)$ be a free matroid on V and let $B = 1$. There are m mappings w_e for $e \in E$ described by: $w_e(v) = 1$ if v is incident to e in G and $w_e(v) = 0$ otherwise.

Clearly, $S \subseteq V$ is a lazy solution iff S is an independent dominating set of G. Moreover, $\sum_{e \in E} w_e(S) = \Delta(G)|S|$. Thus, using Lemma 6, the result follows. □

Using the proof of Theorem 3, we can deduce that any approximation ratio of LAZY STAFF MATROID PROBLEM might depend on parameter m.

Let us now study the generalization of LAZY GREEDY in the context of the LAZY STAFF MATROID PROBLEM with the free matroid. Let LAZY STAFF GREEDY be the algorithm which first renames the elements by non-decreasing sum of their coordinates (ties are broken arbitrarily). At the beginning $I = \emptyset$ and there is a pointer t on the first element. While $t \leq n$, if $I \cup \{t\}$ is a feasible Lazy Staff solution, then $I \leftarrow I \cup \{t\}$, $t \leftarrow t + 1$.

Lemma 7. LAZY STAFF GREEDY *is* $2m$*-approximate on free matroids.*

Proof. let OPT be the value of an optimal solution while APX denotes the value of the solution returned by LAZY STAFF GREEDY. Suppose $OPT \geq \frac{B}{2}$. Since LAZY STAFF GREEDY returns a feasible solution, we get that $APX \leq mB \leq 2mOPT$.

Now suppose $OPT < \frac{B}{2}$. It follows that for every element of the optimum, the sum of its coordinates is at most $OPT < B/2$. Moreover, every element whose sum of its coordinates is at most $B/2$ must be in the optimum (by the maximality constraint). Hence LAZY STAFF GREEDY builds the optimum by taking all elements whose sum of its coordinates is at most $B/2$.

Consider the instance with 3 elements whose weights are $(\frac{B}{2} + \varepsilon, 0, \cdots, 0)$, $(\frac{B}{2} + \varepsilon, \varepsilon, 0, \cdots, 0)$ and $(\frac{B}{2} - \varepsilon, B, \cdots, B)$. LAZY STAFF GREEDY returns a solution which contains elements 1 and 3 while the optimum consists of the second element. The ratio is $mB/(B/2 + 2\varepsilon)$ which tends to $2m$ as ε tends to 0. □

References

1. Arkin, E.M., Bender, M.A., Mitchell, J.S.B., Skiena, S.: The lazy bureaucrat scheduling problem. Inf. Comput. **184** (2003) 129–146
2. Gourvès, L., Monnot, J., Pagourtzis, A.: The lazy bureaucrat problem with common arrivals and deadlines: Approximation and mechanism design. In Gasieniec, L., Wolter, F., eds.: FCT. Volume 8070 of Lecture Notes in Computer Science., Springer (2013) 171–182
3. Gai, L., Zhang, G.: Common deadline lazy bureaucrat scheduling revisited. In Correa, J.R., Hevia, A., Kiwi, M.A., eds.: LATIN. Volume 3887 of Lecture Notes in Computer Science., Springer (2006) 515–523
4. Gai, L., Zhang, G.: On lazy bureaucrat scheduling with common deadlines. J. Comb. Optim. **15** (2008) 191–199
5. Hepner, C., Stein, C.: Minimizing makespan for the lazy bureaucrat problem. In Penttonen, M., Schmidt, E.M., eds.: SWAT. Volume 2368 of Lecture Notes in Computer Science., Springer (2002) 40–50
6. Arkin, E.M., Bender, M.A., Mitchell, J.S.B., Skiena, S.: The lazy bureaucrat scheduling problem. In Dehne, F.K.H.A., Gupta, A., Sack, J.R., Tamassia, R., eds.: WADS. Volume 1663 of Lecture Notes in Computer Science., Springer (1999) 122–133
7. Gai, L., Zhang, G.: Hardness of lazy packing and covering. Oper. Res. Lett. **37** (2009) 89–92
8. Esfahbod, B., Ghodsi, M., Sharifi, A.: Common-deadline lazy bureaucrat scheduling problems. In Dehne, F.K.H.A., Sack, J.R., Smid, M.H.M., eds.: WADS. Volume 2748 of Lecture Notes in Computer Science., Springer (2003) 59–66
9. Camerini, P.M., Vercellis, C.: The matroidal knapsack: A class of (often) well-solvable problems. Operations Research Letters **3** (1984) 157 – 162
10. Halldórsson, M.M.: Approximating the minimum maximal independence number. Information Processing Letters **46** (1993) 169–172
11. Chlebík, M., Chlebíková, J.: Approximation hardness of dominating set problems in bounded degree graphs. Inf. Comput. **206** (2008) 1264–1275
12. Oxley, J.G.: Matroid Theory. Oxford University Press (1992)
13. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer-Verlag Berlin and Heidelberg GmbH & Co. (2003)
14. Brualdi, R.: Comments on bases in different structures. Bull. Austral. Math. Soc. **1** (1969) 161–167
15. Manlove, D.F.: On the algorithmic complexity of twelve covering and independence parameters of graphs. Discrete Applied Mathematics **91** (1999) 155–175
16. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes (extended abstract). In Simon, J., ed.: STOC, ACM (1988) 229–234
17. Alimonti, P., Kann, V.: Some apx-completeness results for cubic graphs. Theor. Comput. Sci. **237** (2000) 123–134
18. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)