The 11th
Hungarian-Japanese Symposium
on Discrete Mathematics and Its Applications

May 27-30, 2019
Tokyo, Japan

Proceedings of the 11th Hungarian-Japanese Symposium
on Discrete Mathematics and Its Applications
May 27-30, 2019, Tokyo, Japan

Editors:

Hiroshi Hirai
Department of Mathematical Informatics
University of Tokyo
hirai@mist.i.u-tokyo.ac.jp

Satoru Iwata
Department of Mathematical Informatics
University of Tokyo
iwata@mist.i.u-tokyo.ac.jp

Shin-ichi Tanigawa
Department of Mathematical Informatics
University of Tokyo
tanigawa@mist.i.u-tokyo.ac.jp

Cover deign: Kazuhiko Shiozaki, 1999

Optimality of the Greedy Algorithm in Greedoids

Dávid Szeszlér*

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Magyar Tudósok krt. 2., Budapest, 1117,
Hungary
szeszler@cs.bme.hu

Abstract: Greedoids were introduced by Korte and Lovász at the beginning of the 1980s as a generalization of matroids. One of the basic motivations of the notion was to extend the theoretical background behind greedy algorithms beyond the well-known results on matroids. Indeed, many well-known algorithms of a greedy nature that cannot be interpreted in a matroid-theoretical context are special cases of the greedy algorithm on greedoids. On the other hand, no general theorem is known that explains the optimality of the greedy algorithm in all these cases. In this paper we generalize a result of Korte and Lovász and thus we obtain a sufficient condition for the optimality of the greedy algorithm that covers a much wider range of known applications than the original one.

Keywords: greedoid, greedy algorithm, matroid

1 Introduction

The term greedoid comes from "a synthetic blending of the words greedy and matroid" [4] which indicates that one of the basic motivations of Korte and Lovász when introducing this notion at the beginning of the 1980s was to extend the theoretical background behind greedy algorithms beyond the well-known results on matroids. Their definition was a result of the observation that in the proofs of various results on matroids subclusiveness (that is, the property that all subsets of independent sets are also independent) is not needed. Therefore the definition of greedoids arises from that of matroids by relaxing this condition (see Definition 1).

Although the research of greedoids was very active until the mid-1990s, the topic seems to have faded away since then. Most of the known results on greedoids are already included in the comprehensive book of Korte, Lovász and Schrader [4] published in 1991. The fact that the notion of greedoids has not gained as much importance within combinatorial optimization as matroids is probably due to the fact that the class of greedoids is much more diverse than that of matroids and classic concepts and results on matroids do not seem to generalize easily to greedoids.

One of the most classic results in the theory of matroids is Edmonds' matroid polytope theorem [1] that gives a description of the polytope spanned by the incidence vectors of all independent sets of a matroid. Most recently in [7] a generalization of an equivalent formulation of the matroid polytope theorem was proved to a class of greedoids, *local forest greedoids*, that contains matroids as well as branching greedoids (that is, edge sets of subtrees of a graph rooted at a given node; see Section 2 for a precise definition). This result was then used to generalize some results of [6] in the field of measuring the reliability of networks by game-theoretical tools.

*Research is supported by the grant No. OTKA 108947 of the Hungarian National Research, Development and Innovation Office (NKFIH).

As it was the case with Edmonds' result, the greedy algorithm turned out to be a fundamental tool for proving the generalization presented in [7] too (although Edmonds' original proof did not seem to extend to the greedoid case). While working on the result of [7] it became apparent that, although many well-known algorithms of a greedy nature are special cases of the greedy algorithm defined on greedoids by Lovász and Korte (see Section 3), no general result is known that would imply the optimality of all these algorithms. Instead, various sufficient conditions on the optimality of the greedy algorithm are known that together cover some of these applications and individual proofs exist for the rest of the cases. Even more surprisingly, it turned out that Lovász and Korte overlooked a detail in [2] and [4] that led them to some false claims concerning this question (see Section 3 for the details). To the best of our knowledge, these mistakes remained hidden for three decades until they were pointed out and corrections were proposed in [7].

The main result of the present paper is a sufficient condition on the optimality of the greedy algorithm in greedoids that generalizes a result of Lovász and Korte [2]. Although this result will still not be able to explain the optimality of the greedy algorithm for all of the above mentioned applications, it will be strong enough to cover important examples beyond the ones covered by the original result.

The structure of this paper is as follows. We summarize the necessary background on greedoids in Section 2. Then preliminaries on the greedy algorithm in greedoids together with various known applications are given in Section 3. Finally, Section 4 is dedicated to the main result of the paper.

2 Preliminaries on Greedoids

Definition 1 *A* greedoid $G = (S, \mathcal{F})$ *is a pair consisting of a finite ground set* S *and a collection of its subsets* $\mathcal{F} \subseteq 2^S$ *such that the following properties are fulfilled:*

(2.1) $\emptyset \in \mathcal{F}$

(2.2) If $X, Y \in \mathcal{F}$ *and* $|X| < |Y|$ *then there exists a* $y \in Y - X$ *such that* $X + y \in \mathcal{F}$.

Members of $\mathcal{F}$ *are called* feasible sets.

Property (2.2) immediately implies that every $X \in \mathcal{F}$ has a *feasible ordering*: the sequence $x_1 x_2 \dots x_k$ is a feasible ordering of X if $X = \{x_1, x_2, \dots, x_k\}$ and $\{x_1, x_2, \dots, x_i\} \in \mathcal{F}$ holds for every $1 \leq i \leq k$. The existence of a feasible ordering, in turn, implies the *accessible property* of greedoids: for every $\emptyset \neq X \in \mathcal{F}$ there exists an $x \in X$ such that $X - x \in \mathcal{F}$.

In this paper, the following notations will be (and have been) used: for a subset $X \subseteq S$ and an element $x \in S$ we will write $X + x$ and $X - x$ instead of $X \cup \{x\}$ and $X - \{x\}$, respectively. Furthermore, given any function $c : S \to \mathbb{R}$ and a subset $X \subseteq S$, $c(X)$ will stand for $\sum\{c(x) : x \in X\}$.

There are many known examples of greedoids beyond matroids and they arise in diverse areas of mathematics, see [4] for an extensive list. For the purposes of this paper, *branching greedoids* will be of importance. Let $H = (V, E_u, E_d)$ be a mixed graph (that is, it can contain both directed and undirected edges) with V, E_u and E_d being its set of nodes, undirected edges and directed edges, respectively. Furthermore, let $r \in V$ be a given root node. The ground set of the branching greedoid on H is $E_u \cup E_d$ and $\mathcal{F}$ consists of all subsets $A \subseteq E_u \cup E_d$ such that disregarding the directions of the arcs in $A \cap E_d$, A is the edge set of a tree containing r and for every path P in A starting in r all edges of $P \cap E_d$ are directed away from r. It is straightforward to check that $G = (E_u \cup E_d, \mathcal{F})$ is indeed a greedoid. G is called an *undirected branching greedoid* or a *directed branching greedoid* if H is an undirected graph (that is, $E_d = \emptyset$) or a directed graph (that is, $E_u = \emptyset$), respectively.

Another simple example is the *poset greedoid*. Let $P = (S, \leq)$ be a partially ordered set. An *ideal* of P is a subset $I \subseteq S$ such that $x, y \in S$, $x \leq y$ and $y \in I$ imply $x \in I$. Then it is easy to to check that if $\mathcal{F}$ consists of all the ideals of P then $G = (S, \mathcal{F})$ is a greedoid. The same can also be expressed in a graph-theoretical setting: if $H = (S, D)$ is an acyclic digraph and $\mathcal{F}$ consists of all subsets of S with in-degree zero then $G = (S, \mathcal{F})$ is the poset greedoid induced by the poset in which $x \leq y$ holds if and only if y is reachable from x via a directed path.

There is an alternative way to define greedoids in terms of languages which is of utmost importance. Denote by S^* the set of all finite sequences $x_1x_2\dots x_k$ of the finite ground set S. Elements of S and S^* are referred to as *letters* and *words*, respectively and a set $\mathcal{L} \subseteq S^*$ is referred to as a *language*. The empty word is denoted by $\emptyset$. The set of letters of a word α is denoted by $\tilde{\alpha}$ and the length of a word α is denoted by $|\alpha|$. $\mathcal{L}$ is a *simple language* if no letter of S appears more than once in any word of $\mathcal{L}$. The concatenation of two words $\alpha, \beta \in S^*$ is simply denoted by $\alpha\beta$. Single letter words are identified with the corresponding element of S (and hence αx denotes the word α followed by the letter x).

Definition 2 *Let S be a finite ground set and $\mathcal{L}$ a simple language on S. Then $G = (S, \mathcal{L})$ is a* greedoid language *if the following properties hold:*

(2.3) $\emptyset \in \mathcal{L}$

(2.4) $\alpha\beta \in \mathcal{L}$ *implies* $\alpha \in \mathcal{L}$

(2.5) If $\alpha, \beta \in \mathcal{L}$ and $|\alpha| > |\beta|$ then there exists an $x \in \tilde{\alpha}$ such that $\beta x \in \mathcal{L}$.

Obviously, the above definition can be viewed as a language equivalent of the definition of matroids. On the other hand, there is a one-to-one correspondence between greedoids and greedoid languages. For a greedoid $G = (S, \mathcal{F})$ let the language $\mathcal{L}(\mathcal{F})$ consist of all feasible orderings of all feasible sets of $\mathcal{F}$. Conversely, for a greedoid language $G = (S, \mathcal{L})$ let $\mathcal{F}(\mathcal{L}) = \{\tilde{\alpha} : \alpha \in \mathcal{L}\}$. Then it is easy to show (see [4, Theorem IV.1.2]) that for every greedoid $(S, \mathcal{F})$, $(S, \mathcal{L}(\mathcal{F}))$ is a greedoid language and conversely, for every greedoid language $(S, \mathcal{L})$, $(S, \mathcal{F}(\mathcal{L}))$ is a greedoid; moreover, in both cases we get the unique greedoid language and greedoid, respectively, for which $\mathcal{F}(\mathcal{L}(\mathcal{F})) = \mathcal{F}$ and $\mathcal{L}(\mathcal{F}(\mathcal{L})) = \mathcal{L}$ holds. This relation between greedoids and greedoid languages justifies the above claim that greedoid languages can be thought of as an alternative way to define greedoids.

Some of the well-known terminology on matroids can be applied to greedoids without any modification. In particular, a *base* of a greedoid $G = (S, \mathcal{F})$ is a feasible set $X \in \mathcal{F}$ of maximum size. This, by property (2.2), is equivalent to saying that $X + y \notin \mathcal{F}$ for every $y \in S - X$. Analogously, a *basic word* of a greedoid language $G = (S, \mathcal{L})$ is a word $\alpha \in \mathcal{L}$ of maximum length; or, equivalently by property (2.5), a word $\alpha \in \mathcal{L}$ for which $\alpha x \notin \mathcal{L}$ for every $x \in S - \tilde{\alpha}$. Both the set of bases and the set of basic words will be denoted by $\mathcal{B}$.

3 Preliminaries on the Greedy Algorithm on Greedoids

Let $G = (S, \mathcal{L})$ be a greedoid language and $w : \mathcal{L} \to \mathbb{R}$ an objective function. Assume that we are interested in finding a basic word $\beta \in \mathcal{B}$ that minimizes $w(\beta)$ across all basic words of G.

For every $\alpha \in \mathcal{L}$ the *set of continuations* of α is defined as $\Gamma(\alpha) = \{x \in S - \tilde{\alpha} : \alpha x \in \mathcal{L}\}$. Then the *greedoid greedy algorithm* for the above problem can be described as follows [2, 4]:

Step 1. Set $\alpha = \emptyset$.

Step 2. If $\Gamma(\alpha) = \emptyset$ then stop and output α.

Step 3. Choose an $x \in \Gamma(\alpha)$ such that $w(\alpha x) \leq w(\alpha y)$ for every $y \in \Gamma(\alpha)$.

Step 4. Replace α by αx and continue at *Step 2*.

Obviously, if the task is to maximize $w(\beta)$ across all basic words then, since this is equivalent to minimizing $-w(\beta)$, $w(\alpha x) \geq w(\alpha y)$ is to be required for every $y \in \Gamma(\alpha)$ in Step 3.

Many of the well-known, elementary algorithms in graph theory fall under this framework as shown by the following examples.

Example 3 (Matroid greedy algorithm) *If M is a matroid and w is linear (meaning that $w(A) = c(A)$ for some weight function $c : S \to \mathbb{R}$) then the greedoid greedy algorithm is nothing but the well-known greedy algorithm on matroids. In particular, we get Kruskal's algorithm for finding a minimum weight spanning tree in case of the cycle matroid.*

Example 4 (Prim's algorithm) *Let G be the branching greedoid of the undirected, connected graph H and w a linear objective function. Then the greedoid greedy algorithm translates to Prim's well-known algorithm for finding a minimum weight spanning tree. (Note that this algorithm cannot be interpreted in a matroid-theoretical context.)*

Example 5 (Dijkstra's shortest path algorithm) *Let G be the branching greedoid of the mixed graph $H = (V, E_u, E_d)$ with root node r and let $c : E_u \cup E_d \to \mathbb{R}^+$ be a non-negative valued weight function. For every feasible set A and $e \in A$ let P_e^A denote the unique path in A starting at r and ending in e. Then let $w(A) = \sum\{c(P_e^A) : e \in A\}$ for every $A \in \mathcal{F}$. Korte and Lovász observed in [2] that in this case the greedoid greedy algorithm for minimizing $w(B)$ translates to Dijkstra's well-known shortest path algorithm. Indeed, Dijkstra's algorithm constructs a spanning tree on the set of nodes reachable from r such that the unique path from r to every other node in this tree is a shortest path and hence it clearly minimizes w.*

Example 6 (Dijkstra's widest path algorithm) *Given a graph with weights on its edges, the widest path problem is the problem of finding a path between two given vertices that maximizes the weight of the minimum-weight edge on the path. It is well-known (and it seems to belong to graph theory folklore) that a trivial modification of Dijkstra's shortest path algorithm solves this problem too. This is again a special case of the greedoid greedy algorithm: let $c : E_u \cup E_d \to \mathbb{R}$ be a (real valued) weight function on the edge set of the mixed graph $H = (V, E_u, E_d)$ with root node r, let $\mu(P_e^A) = \min\{c(z) : z \in P_e^A\}$ for every path of H starting at r and let $w(A) = \sum\{\mu(P_e^A) : e \in A\}$ for every feasible set A of the branching greedoid of H. Then, analogously to the above example, the greedoid greedy algorithm for maximizing $w(B)$ translates to the above mentioned modified version of Dijkstra's algorithm (and hence it is optimal) and it constructs a spanning tree on the set of nodes reachable from r such that the unique path from r to every other node in this tree is a widest path.*

Example 7 (Lawler's single machine scheduling algorithm) *Let $D = (V, A)$ be an acyclic digraph whose vertices represent jobs to be scheduled on a single machine (with no interruptions) and arcs of D represent precedence constraints to be respected by the schedule. Furthermore, a processing time $a(v) \in \mathbb{N}$ is also given for every job $v \in V$. Finally, a monotone non-decreasing cost function $c_v : \{0, \dots, N\} \to \mathbb{R}$ is also given for every job $v \in V$, where $N = \sum_{v \in V} a(v)$ such that $c_v(t)$ represents the cost incurred by job v if it is completed at time t. The problem is to find a schedule (that is, a topological ordering of the jobs with respect to D) such that the maximum of the costs incurred is minimized. Lawler [5] gave a simple greedy algorithm for this problem: it builds up the schedule in a reverse order always choosing out of the currently possible jobs one with the lowest cost at the current completion time. As it was pointed out in [2], this algorithm is also a special case of the greedoid greedy algorithm if the underlying greedoid is the poset greedoid induced by the digraph obtained from D by reversing all its arcs. (More precisely: each legal running of Lawler's algorithm is a legal running of the greedoid greedy algorithm, but not vice versa. On the other hand, the optimality of both is easily proved by the method of [5].)*

Although the greedy algorithm is optimal in the above examples, it is obviously not to be expected that this is true in general. The first sufficient condition on the optimality of the greedy algorithm was given by Korte and Lovász in [2].

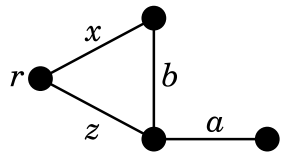
Theorem 8 ([2],[4, Theorem XI.1.3]) *Let $G = (S, \mathcal{L})$ be a greedoid language and $w : \mathcal{L} \to \mathbb{R}$ an objective function. Assume that for every $\alpha x \in \mathcal{L}$ such that $w(\alpha x) \leq w(\alpha y)$ for every $y \in \Gamma(\alpha)$ the following conditions hold:*

(3.1) $w(\alpha\beta x\gamma) \leq w(\alpha\beta z\gamma)$, *if* $\alpha\beta x\gamma, \alpha\beta z\gamma \in \mathcal{L}$;

(3.2) $w(\alpha x\beta z\gamma) \leq w(\alpha z\beta x\gamma)$, *if* $\alpha x\beta z\gamma, \alpha z\beta x\gamma \in \mathcal{L}$.

Then the greedoid greedy algorithm finds a basic word of minimum weight with respect to w.

Unfortunately, Theorem 8 does not imply the optimality of the greedy algorithm for all of the above listed examples; in fact, it is only Example 3 that it completely covers. Indeed, it is easy to verify that the conditions of Theorem 8 are fulfilled by linear objective functions on matroids and hence Theorem 8 is a generalization of the Edmonds-Rado theorem. However, this verification strongly relies on the property of matroids that all subsets of independent sets are also independent and therefore it is not a surprise that (3.1) is violated in case of Example 4 (as shown by trivial examples). In case of Examples 5 and 6 it is easy to verify that the conditions of Theorem 8 are fulfilled if the underlying graph is directed (that is, the greedoid is a directed branching greedoid). However, although the same is also claimed for Example 5 in the undirected case in [4, page 156], condition (3.1) is not necessarily fulfilled either for Example 5 or for 6 in case of undirected graphs (that is, undirected branching greedoids) as shown by the following simple examples. In case of Example 5 consider the graph of Figure 1 with edge weights $c_1(e)$; then although x is the best continuation of $\emptyset$, $11 = w(xba) > w(zba) = 10$, hence w violates (3.1) with $\alpha = \beta = \emptyset$ and $\gamma = ba$. Similarly, the same graph with edge weights $c_2(e)$ shows that $(-w)$ violates (3.1) in case of Example 6 since again x is the best continuation of $\emptyset$, but $6 = w(xba) < w(zba) = 7$.



e	x	z	a	b
$c_1(e)$	1	2	0	4
$c_2(e)$	4	3	3	1

Figure 1:

Finally, in case of Example 7 it is again easy to verify that conditions (3.1) and (3.2) are fulfilled if all processing times $a(v)$ are equal (and hence the cost incurred by a job only depends on its position in the seqence and not on the processing times of previously completed jobs). Therefore Theorem 8 implies the optimality of the greedy algorithm in this special case. However, as opposed to what is claimed in [2], the same is not true in the general case. To simplify the description of a counterexample, we define the "reverse-Lawler scheduling problem": it is identical to the one described in Example 7 with the difference being that the cost functions c_v are monotone non-increasing and $c_v(t)$ represents the cost incurred by job v if it is started at time t. Obviously, this problem is equivalent to the original one by reversing all possible schedules, however, in this version the basic words of the corresponding poset greedoid are identical to the possible schedules (and not the reverses of those). Then consider the following example: let $S = \{x, y, z\}$, let D be the empty graph (meaning that there are no precedence constraints), let $a(x) = a(y) = 1$ and $a(z) = 2$, and finally let the cost functions c_v be as in the table below.

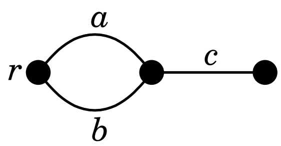
t	0	1	2	3	4
$c_x(t)$	0	0	0	0	0
$c_y(t)$	2	2	1	0	0
$c_z(t)$	1	0	0	0	0

Then x is the best continuation of $\emptyset$, but $w(xy) = \max\{0, 2\} = 2$ and $w(zy) = \max\{1, 1\} = 1$ which shows that (3.1) is violated with $\alpha = \beta = \emptyset$ and $\gamma = y$.

As it was the case in Examples 3–6, in most applications the objective function only depends on the feasible sets themselves and not on their orderings; in other words, it is a greedoid $G = (S, \mathcal{F})$ that the objective function $w : \mathcal{F} \to \mathbb{R}$ is defined on. Therefore one would want to formulate the corresponding corollary of Theorem 8. Obviously, (3.2) is automatically fulfilled in these cases, however, it is not at all straightforward to specialize (3.1) to such objective functions. Both in [2] and [4, Chapter XI] it is claimed that (3.1) is equivalent to the following for objective functions $w : \mathcal{F} \to \mathbb{R}$:

(3.3) If $A, B, A + x, B + x \in \mathcal{F}$ hold for some sets $A \subseteq B$ and $x \in S - B$, and $w(A + x) \leq w(A + y)$ for every $y \in \Gamma(A)$ then $w(B + x) \leq w(B + z)$ for every $z \in \Gamma(B)$.

However, as it was pointed out in [7], this reformulation clearly disregards the fact that $B = \tilde{\alpha} \cup \tilde{\beta} \cup \tilde{\gamma}$ need not be a feasible set. In actual fact, (3.3) does not guarantee the optimality of the greedy algorithm as shown by the trivial example of Figure 2: consider the undirected branching greedoid of the graph on the left hand side and let the objective function be defined as in the table on the right hand side. It is easy to check that (3.3) is fulfilled, however, the greedy algorithm gives $\{a, c\}$ instead of $\{b, c\}$. On the other hand, (3.1) is clearly violated with $\alpha = \beta = \emptyset$, $\gamma = c$, $x = a$ and $z = b$: a is the best continuation of $\emptyset$ but $w(ac) > w(bc)$.



X	$\emptyset$	$\{a\}$	$\{b\}$	$\{a, c\}$	$\{b, c\}$
$w(X)$	0	3	4	2	1

Figure 2:

Unfortunately, as innocuous as the above mistake might look, it led the authors of [4] to some further false claims (including the above mentioned one that Example 5 is covered by Theorem 8 for all graphs); see [7] for the details.

In case of linear objective functions, the following was proved in [3].

Theorem 9 ([3],[4, Theorem XI.2.2]) *Let $G = (S, \mathcal{F})$ be a greedoid. Then the greedoid greedy algorithm finds a minimum weight base for every linear objective function $c : S \to \mathbb{R}$ if and only if the following condition holds:*

(3.4) If $A, A + x \in \mathcal{F}$, $B \in \mathcal{B}$ hold for some sets $A \subseteq B$ and $x \in S - B$ then there exists a $y \in B - A$ such that $A + y \in \mathcal{F}$ and $B - y + x \in \mathcal{B}$.

It is easy to verify that (3.4) is fulfilled by the undirected branching greedoid and hence Theorem 9 implies the optimality of the greedy algorithm in case of Example 4 above.

A generalization of the sufficiency of (3.4) to arbitrary order-independent objective functions was given in [7].

Theorem 10 ([7]) *Let $G = (S, \mathcal{F})$ be a greedoid and $w : \mathcal{F} \to \mathbb{R}$ an objective function that fulfills the following property:*

(3.5) If for some $A \subseteq B$, $A, A + x \in \mathcal{F}$, $B \in \mathcal{B}$ and $x \in S - B$ it holds that $w(A + x) \leq w(A + u)$ for every $u \in \Gamma(A)$ then there exists a $y \in B - A$ such that $B - y + x \in \mathcal{B}$ and $w(B - y + x) \leq w(B)$.

Then the greedy algorithm gives a minimum base with respect to w.

A certain necessity of (3.5) was also proved in [7]: if (3.5) is violated then the greedy algorithm can give a suboptimal base in a minor of the greedoid; see [7] for the details. So Theorem 10 seems to be the best possible that can be said about the optimality of the greedoid greedy algorithm in case of order-independent objective functions (in spite of the fact that its proof is short and simple and it is an adaptation of the proof of Theorem 9). It can also be easily verified that it implies the optimality of the greedy algorithm in all Examples 3-6 above.

4 Main Result

We aim at proving a generalization of Theorem 8: we will give a sufficient condition on the optimality of the greedoid greedy algorithm for (possibly) order-dependent objective functions that is weaker than requiring (3.1) and (3.2) and which, besides other examples, will completely cover Example 7 above. Claiming the theorem needs some preparation.

Definition 11 *Let $G = (S, \mathcal{L})$ be a greedoid language, $\alpha \in \mathcal{L}$, $x \in S$, $y \in \tilde{\alpha}$ and $\alpha = \alpha_0 y \alpha_1$. Then y is said to be an* entry point of x in α *if $\alpha_0 x \in \mathcal{L}$.*

Note that $x \in \tilde{\alpha}$ is possible and in that case x is an entry point of itself in α by (2.4).

Definition 12 *Let $G = (S, \mathcal{L})$ be a greedoid language, $\alpha \in \mathcal{L}$ and $x \in S$ that has at least one entry point in α. Assume that $y_1, y_2, \ldots, y_r \in \tilde{\alpha}$ is the list of all entry points of x in α appearing in this order and for every $1 \leq i \leq r$ let $\alpha_x^{y_i}$ denote the word obtained from α by replacing y_i with x and y_j with y_{j-1} for every $i < j \leq r$. That is, if $\alpha = \alpha_0 y_1 \alpha_1 y_2 \alpha_2 \ldots y_r \alpha_r$ then*

$$\alpha_x^{y_i} = \alpha_0 y_1 \alpha_1 \ldots y_{i-1} \alpha_{i-1} x \alpha_i y_i \alpha_{i+1} y_{i+1} \ldots y_{r-1} \alpha_r.$$

Obviously, if $x \in \tilde{\alpha}$ then $x = y_r$ is the last entry point of itself in α and $\alpha_x^x = \alpha$. The following lemma is a generalization of a lemma proved in [2]; its proof is also a refinement of the one given in [2], but it is simpler than that.

Lemma 13 *Assume that $G = (S, \mathcal{L})$ is a greedoid language, $\alpha \in \mathcal{L}$, $x \in S$ and $y \in \tilde{\alpha}$ is an entry point of x in α. Then $\alpha_x^y \in \mathcal{L}$.*

Proof: We proceed by induction on $|\alpha|$. If $|\alpha| = 1$ then the claim is trivial, so assume $|\alpha| > 1$. Let the last letter of α be z, $\alpha = \bar{\alpha} z$ and let u denote the last entry point of x in $\bar{\alpha}$ if it exists.

Assume first that z is an entry point of x in α. Then $\alpha_x^z = \bar{\alpha} x$, so $\alpha_x^z \in \mathcal{L}$ (since z is an entry point). For all other entry points $y \neq z$ of x in α, $\alpha_x^y = \bar{\alpha}_x^y u$. We have $\bar{\alpha}_x^y$ by induction. Then applying (2.5) on $\bar{\alpha}_x^y$ and $\bar{\alpha} x$ we get $\alpha_x^y = \bar{\alpha}_x^y u \in \mathcal{L}$ as claimed.

So assume now that z is not an entry point of x, that is, $\bar{\alpha} x \notin \mathcal{L}$. We again have $\bar{\alpha}_x^y$ by induction. Applying (2.5) on $\bar{\alpha}_x^y$ and α we get that either $\bar{\alpha}_x^y z \in \mathcal{L}$ or $\bar{\alpha}_x^y u \in \mathcal{L}$. The latter is clearly impossible if $x \in \tilde{\alpha}$ (since in that case $u = x$ must be true). However, even in the $x \notin \tilde{\alpha}$ case if $\bar{\alpha}_x^y u \in \mathcal{L}$ were true then applying (2.5) on $\bar{\alpha}$ and $\bar{\alpha}_x^y u$ would imply $\bar{\alpha} x \in \mathcal{L}$ contradicting our assumption. Therefore $\alpha_x^y = \bar{\alpha}_x^y z \in \mathcal{L}$ as claimed. □

Now we are ready for the main result of this paper. Again, its proof follows that of Theorem 8, but it is shorter and simpler than that.

Theorem 14 *Let $G = (S, \mathcal{L})$ be a greedoid language and $w : \mathcal{L} \to \mathbb{R}$ an objective function. Assume that the following condition holds:*

(4.1) If $\alpha x \in \mathcal{L}$ such that $w(\alpha x) \leq w(\alpha y)$ for every $y \in \Gamma(\alpha)$ and $\gamma = \alpha z \beta \in \mathcal{B}$ is a basic word then $w(\gamma_x^z) \leq w(\gamma)$.

Then the greedoid greedy algorithm finds a basic word of minimum weight with respect to w.

Proof: Assume by way of contradiction that the greedy algorithm gives the basic word δ that is not optimal with respect to w. Let γ be a minimum weight basic word with respect to w and choose γ such that its common prefix with δ is the longest possible among all optimal basic words. Let this common prefix be α and let $\delta = \alpha x \delta'$ and $\gamma = \alpha z \beta$. Then $w(\alpha x) \leq w(\alpha y)$ for every $y \in \Gamma(\alpha)$ follows from the operation of the greedy algorithm. Therefore $w(\gamma_x^z) \leq w(\gamma)$ holds by (4.1). Hence γ_x^z is also a minimum weight basic word with respect to w, but it has a longer common prefix with δ than γ contradicting the choice of γ. □

Proposition 15 *Conditions (3.1) and (3.2) together imply condition (4.1) (and therefore Theorem 14 is a generalization of Theorem 8).*

Proof: Let $\gamma = \alpha z \beta \in \mathcal{B}$ be a basic word and $\alpha x \in \mathcal{L}$ such that $w(\alpha x) \leq w(\alpha y)$ for every $y \in \Gamma(\alpha)$. Let the entry points of x in γ that do not belong to $\tilde{\alpha}$ be $z = y_1, y_2, \dots, y_r$ in this order. Then (3.1) implies $w(\gamma) \geq w(\gamma_x^{y_r})$. Furthermore, applying the combination of (3.1) and (3.2) consecutively $r-1$ times we get $w(\gamma_x^{y_r}) \geq w(\gamma_x^{y_{r-1}}) \geq \ldots \geq w(\gamma_x^{y_2}) \geq w(\gamma_x^{z})$. These together give $w(\gamma) \geq w(\gamma_x^z)$ as claimed. □

The proof of the following proposition follows the proof given in [5].

Proposition 16 *Condition (4.1) is fulfilled by the objective function of Example 7 (and therefore the optimality of Lawler's scheduling algorithm follows from Theorem 14).*

Proof: For the sake of simplicity, we will give the proof for the equivalent "reverse-Lawler" problem described after Theorem 8. So assume that the set of jobs V with the precedence digraph $D = (V, A)$, processing times $a(v) \in \mathbb{N}$ and non-increasing cost functions c_v are given.

Assume that $\gamma = \alpha z \beta \in \mathcal{B}$ is a basic word of the poset greedoid $(V, \mathcal{L})$, that is, a topological ordering of all the jobs. Assume further that x is the best continuation of α. If $x = z$ then there is nothing to prove by $\gamma_x^x = \gamma$, so assume the opposite. Then $x \in \tilde{\beta}$ (since γ lists all the jobs) so let $\beta = \beta_1 x \beta_2$. Obviously, every $v \in \tilde{\beta}_1$ is an entry point of x in γ by $\alpha x, \gamma \in \mathcal{L}$ and the definition of the poset greedoid. Therefore $\gamma_x^z = \alpha x z \beta_1 \beta_2$. This shows that for every job $v \neq x$ the starting time of v in γ_x^z is at least as big as in γ and hence the cost incurred by v in γ_x^z is at most the one as in γ. Let $t = \sum_{v \in \tilde{\alpha}} a(v)$. Then $w(\alpha x) \leq w(\alpha z)$ implies that either $c_x(t) \leq c_z(t)$ or there is a job $v \in \tilde{\alpha}$ such that the cost incurred by v in γ is at least as big as $c_x(t)$. In both cases we get $w(\gamma_x^z) \leq w(\gamma)$ as claimed. □

It is also worth noting that, besides Example 7, Theorem 14 implies the optimality of the greedy algorithm in all the examples listed in Section 3 except for Example 4. However, it remains an open problem to find a common generalization of Theorems 10 and 14 that covers all of these examples.

References

[1] J. Edmonds, Matroids and the greedy algorithm, *Mathematical Programming* **1(1)**, pp. 127-136 (1971).

[2] B. Korte and L. Lovász, Greedoids – a structural framework for the greedy algorithm, *in: W. Pulleyblank (ed.), Progress in combinatorial optimization, Academic Press, London*, pp. 221-243 (1984).

[3] B. Korte and L. Lovász, Greedoids And Linear Objective Functions, *SIAM Journal on Algebraic Discrete Methods* **5(2)**, pp. 229-238 (1984).

[4] B. Korte, L. Lovász and R. Schrader, *Greedoids*, 211 pages, Springer-Verlag, Berlin (1991).

[5] Lawler E. L., Optimal sequencing of a single machine subject to precedence constraints, *Management science* **19(5)**, pp. 544-546 (1973).

[6] D. Szeszlér, Security games on matroids, *Mathematical Programming* **161(1)**, pp. 347-364, (2017).

[7] D. Szeszlér, A Generalization of the Matroid Polytope Theorem to Local Forest Greedoids, submitted. https://arxiv.org/abs/1811.04690