An Algorithmic Characterization of Antimatroids

by

E. Andrew Boyd

Abstract

In an article entitled "Optimal sequencing of a single machine subject to precedence constraints," E. L. Lawler presented a now classical minmax result for job scheduling. In essence, Lawler's proof demonstrated that the properties of partially ordered sets were sufficient to solve the posed scheduling problem. These properties are, in fact, common to a more general class of combinatorial structures known as antimatroids, which have recently received considerable attention in the literature. It is demonstrated that the properties of antimatroids are not only sufficient but necessary to solve the scheduling problem posed by Lawler, thus yielding an algorithmic characterization of antimatroids. Examples of problems solvable by the general result are provided.
0.1 Introduction

In an article entitled "Optimal sequencing of a single machine subject to precedence constraints," E. L. Lawler showed how to solve the $1|\text{prec}|f_{\max}$ scheduling problem using a variant of the greedy algorithm. In his proof, Lawler took advantage of the fact that the underlying constraints of the problem were the constraints imposed by a partial order on the jobs to be scheduled. In actuality, the properties used by Lawler in his proof are common to a more general class of combinatorial structures known as antimatroids.

Antimatroids were apparently first considered by Edelman [1980] and by Jamison-Waldner [1982]. Korte and Lovász [1982] studied these structures from a different perspective under the names alternative precedence structures and upper interval greedoids. Antimatroids have recently received considerable attention in the literature (e.g., Korte and Lovász [1984b], [1985a], [1985b]). There are a variety of reasons for this attention. As pointed out by Korte and Lovász [1982], the class of antimatroids includes many interesting combinatorial structures within its realm. In addition, antimatroids are closely related to matroids in that both can be defined by a very similar set of axioms, the only difference being an exchange axiom for matroids but an anti-exchange axiom for antimatroids [Korte and Lovász 1984b]. This close relationship to matroids has provided a fruitful combinatorial structure that is general enough to be interesting while still maintaining sufficient structure to be amenable to proofs.

This paper provides an algorithmic characterization of antimatroids that helps to provide further insight into the structure and algorithmic relevance of antimatroids. Section 2 introduces basic information about antimatroids and greedoids that will be needed for the following sections. In Section 3 the $1|\text{prec}|f_{\max}$ scheduling problem is presented in a more abstract form that allows it to be defined on an arbitrary combinatorial structure and in Section 4 necessary properties of truncated antimatroids are introduced. The main result is proved in Section 5, where antimatroids are characterized as exactly those combinatorial structures for which the associated scheduling problem is certain to succumb to the greedy algorithm. The paper is concluded with some examples of special cases of the scheduling problem.
0.2 Preliminaries

We begin by presenting tools from the theory of greedoids necessary for the development in the following sections. For a more detailed treatment, the reader is directed to Korte and Lovász [1982] [1983c] or to Björner and Ziegler [1986].

Let $\mathcal{F} \subseteq 2^E$ be a set system defined on the finite ground set $E$. One of the many equivalent axiomatizations of greedoids is the following.

**Definition:** A *greedoid* is a nonempty set system $(E, \mathcal{F})$ satisfying the following two properties.

1. $\emptyset \in \mathcal{F}$
2. if $X, Y \in \mathcal{F}$ and $|X| > |Y|$ then there exists an $x \in X - Y$ such that $Y \cup \{x\} \in \mathcal{F}$

An alternative definition makes use of the well-developed theory of formal languages. Given a finite alphabet $E$, a language $\mathcal{L}$ is a nonempty collection of words consisting of letters in $E$. Words will be denoted by the lower case greek letters $\alpha, \beta$, and $\gamma$, or by a specified sequence of elements in $E$ such as $\alpha = x_1 \ldots x_k$. The concatenation of two words $\alpha$ and $\beta$ will be denoted $\alpha \beta$, and the length of a word $\alpha$ by $|\alpha|$. The $k^{th}$ element of a word $\alpha$ will be denoted by $\alpha(k)$, and $\alpha_k$ will be used to denote a word of length $k$ or the $k^{th}$ subword of $\alpha$. The empty string will be denoted by $\varepsilon$ or by $\alpha_0$ when referring to a subword of $\alpha$ of length 0. The set of distinct elements in a word $\alpha$ will be denoted $\alpha^*$. Words contained in a language $\mathcal{L}$ are feasible. A word $\alpha_k$ is accessible if $\alpha_i \in \mathcal{L}$ for $i = 0, \ldots, k$, otherwise it is inaccessible. A language $\mathcal{L}$ is accessible if every word in $\mathcal{L}$ is accessible. A word $\alpha$ is simple if no letter appears more than once, or equivalently, if $|\alpha| = |\alpha^*|$. A language $\mathcal{L}$ is simple if every word in $\mathcal{L}$ is simple. The finite set of all simple words defined by an alphabet $E$ will be denoted by $E^0$. Using formal languages, a *greedoid language* can be defined as follows.

**Definition:** A *greedoid language* is a simple language $(E, \mathcal{L})$ satisfying the following two properties.
1. if $\alpha \in \mathcal{L}$ and $\alpha = \beta \gamma$, then $\beta \in \mathcal{L}$

2. if $\alpha, \beta \in \mathcal{L}$ and $|\alpha| > |\beta|$, then there exists an $x \in \alpha$ such that $\beta x \in \mathcal{L}$

The relationship between greedoids and greedoid languages is expressed in a theorem proved by Korte and Lovász [1982].

**Theorem 1:** If $(E, \mathcal{L})$ is a greedoid language then

$$F(\mathcal{L}) = \{\alpha^* : \alpha \in \mathcal{L}\}$$

is a greedoid $(E, F(\mathcal{L}))$. Conversely, if $(E, \mathcal{F})$ is a greedoid, then

$$L(\mathcal{F}) = \{x_1 \ldots x_k \in E^0 : \{x_1, \ldots, x_j\} \in \mathcal{F} \text{ for } j = 0, \ldots, k\}$$

is a greedoid language $(E, L(\mathcal{F}))$. Further, $L(F(\mathcal{L})) = \mathcal{L}$ and $F(L(\mathcal{F})) = \mathcal{F}$.

The immediate implication of Theorem 1 is that greedoids can be considered equivalently as set systems or as simple languages. Henceforth, the term greedoid will be used for both greedoids and greedoid languages and the definition used will depend upon the development at hand.

The $k$-truncation or simply truncation of a simple language $(E, \mathcal{L})$ is the simple language defined by

$$\mathcal{L}_k = \{\alpha \in \mathcal{L} : |\alpha| \leq k\}$$

while the restriction of a simple language $(E, \mathcal{L})$ to a set $A \subseteq E$ is the simple language defined by

$$\mathcal{L}|_A = \{\alpha \in \mathcal{L} : \alpha^* \subseteq A\}$$

Clearly, if $(E, \mathcal{L})$ is a greedoid then the $k$-truncation and restriction of $(E, \mathcal{L})$ are also greedoids.

The rank of a set $A \subseteq E$ is defined as

$$\rho(A) = \max\{|\alpha| : \alpha \in \mathcal{L}|_A\}$$
The rank of a simple language \((E, \mathcal{L})\), while properly denoted \(\rho(E)\), will be denoted \(\rho(\mathcal{L})\).

The greedy algorithm has a natural definition for greedoids. The following formal definition is provided for reference.

**Definition:** The Greedy Algorithm.

Let \((E, \mathcal{L})\) be a simple language with an associated function \(W : \mathcal{L} \rightarrow \mathbb{R}\).

Let \(\alpha\) initially be the empty word.

Choose \(x \in E - \alpha^*\) such that

1. \(\alpha x \in \mathcal{L}\)
2. \(W(\alpha x) \leq W(\alpha y)\) for all \(y\) such that \(\alpha y \in \mathcal{L}\)

Let \(\alpha = \alpha x\) and repeat until \(\alpha\) can no longer be augmented.

### 0.3 Problem

In order to extend Lawler’s result and characterize antimatroids, it is necessary to generalize the \(1|\text{prec}|f_{\text{max}}\) scheduling problem to include arbitrary combinatorial structures. The following definitions provide the necessary generalization.

**Definition:** Let \(E\) be a finite ground set and let \(f : E \times 2^E \rightarrow \mathbb{R}\). A maximum nesting function is a function of the form

\[
W(x_1, \ldots, x_k) = \max \{f(x_1, \{x_1\}), \ldots, f(x_k, \{x_1, \ldots, x_k\})\}
\]

A maximum nesting function will be called \(f\)-monotone if \(f(x, A) \leq f(x, B)\) whenever \(B \subseteq A\).

The optimization problem of interest can now be defined as follows.
**Definition:** The Minmax Nesting Problem. Given a simple language \((E, \mathcal{L})\) with an \(f\)-monotone maximum nesting function \(W\) and a non-negative integer \(k \leq \rho(\mathcal{L})\), find \(\alpha_k \in \mathcal{L}\) such that

\[
W(\alpha_k) = \min \{W(\beta_k) : \beta_k \in \mathcal{L}\}
\]

As an example, consider the problem of scheduling a set of jobs \(E\) on a single machine subject to precedence constraints. A simple language \((E, \mathcal{L})\) is defined by the set of partial schedules of these jobs satisfying the given precedence constraints. In fact, a simple language defined in this way is a greedoid known as a schedule or poset greedoid [Korte and Lovász 1982].

Assume each job \(x\) has an associated processing time \(p_x\) and a nonincreasing cost function \(c_x(t)\) giving the cost of completing job \(x\) at time \(t\), and define \(f : E \times 2^E \to \mathbb{R}\) by

\[
f(x, A) = c_x(\sum_{y \in A \cup \{x\}} p_y)
\]

Clearly, \(f(x, A) \leq f(x, B)\) if \(B \subseteq A\). Thus, the maximum nesting function defined by \(f\) is \(f\)-monotone. The maximum nesting problem in this instance is to determine a partial schedule of \(k\) jobs that minimizes the maximum cost incurred by any job in the partial schedule.

### 0.4 Truncated Antimatroids

The \(f\)-monotone maximum nesting problem presented in the previous section is intimately related with antimatroids or, more specifically, with truncated antimatroids. There are many alternative definitions of antimatroids all of which provide different insights into this combinatorial structure. A common definition that will prove useful for the purposes of this paper is the following.

**Definition:** A simple language \((E, \mathcal{L})\) is said to have the upper interval property if \(\alpha, \alpha\beta, \alpha x \in \mathcal{L}\) with \(x \not\in \beta^* \Rightarrow \alpha\beta x \in \mathcal{L}\).

**Definition:** A greedoid \((E, \mathcal{L})\) with the upper interval property is an antimatroid.
From an algorithmic perspective a similar property is the following.

**Definition:** A simple language \((E, \mathcal{L})\) is said to have the *truncated upper interval property* if \(\alpha, \alpha \beta, \alpha x, \in \mathcal{L}\) with \(x \notin \beta^*\) and \(|\alpha \beta| < \rho(\mathcal{L}) \Rightarrow \alpha \beta x \in \mathcal{L}\).

Note that the truncated upper interval property is a relaxation of the upper interval property. In fact, the above definitions suggest that the operation of truncation provides the formal link between greedoids with the truncated upper interval property and antimatroids. Certainly, the truncation of any antimatroid has the truncated upper interval property. But even further, any greedoid with the truncated upper interval property defines in a natural way an antimatroid of which it is a truncation, as the following proofs demonstrate.

**Observation:** If a set system \((E, \mathcal{F})\) is a greedoid, then the truncated upper interval property is equivalent to

\[X, Y \in \mathcal{F}, X \subseteq Y, X \cup \{x\} \in \mathcal{F}, |Y| < \rho(\mathcal{F}) \Rightarrow Y \cup \{x\} \in \mathcal{F}\]

**Proposition 1:** A greedoid \((E, \mathcal{F})\) has the truncated upper interval property if and only if

\[X, Y \in \mathcal{F}, |X \cup Y| \leq \rho(\mathcal{F}) \Rightarrow X \cup Y \in \mathcal{F}\]

**Proof:** *(if):* Consider any sets \(X, Y \in \mathcal{F}\) such that \(X \subseteq Y, X \cup \{x\} \in \mathcal{F},\) and \(|Y| < \rho(\mathcal{F})\). Clearly, \(|Y \cup (X \cup \{x\})| \leq \rho(\mathcal{F})\), and thus by assumption \(Y \cup (X \cup \{x\}) = Y \cup \{x\} \in \mathcal{F}\).

*(only if):* Assume \(\mathcal{F}\) has the truncated upper interval property and consider two sets \(X, Y \in \mathcal{F}\) with \(|X \cup Y| \leq \rho(\mathcal{F})\). Let \(y_1...y_k\) be any ordering of \(Y\) such that every subword is feasible in \(\mathcal{F}\), and let \(Y_i = (y_1...y_i)^*\) with \(Y_0 = \emptyset\). Let \(j\) be the smallest index such that \(X \cup Y \neq X \cup Y_{j-1}\) but \(X \cup Y = X \cup Y_j\); that is, \(\{y_{j+1}, ..., y_k\} \subseteq X\). Clearly, \(Y_i \subseteq X \cup Y_i, Y_i \cup \{y_{i+1}\} \in \mathcal{F}\), and for \(i < j, |X \cup Y_i| < \rho(\mathcal{F})\). Since \(X \cup Y_0 \in \mathcal{F}\) and \(\mathcal{F}\) has the truncated upper interval property by assumption, it follows by induction that \(X \cup Y_i \in \mathcal{F}\) for \(i = 0, ..., j\). In particular, \(X \cup Y_j = X \cup Y \in \mathcal{F}\).
Proposition 2: Given a greedoid \((E, \mathcal{F})\) of rank \(m\) with the truncated upper interval property, the set system \((E, \mathcal{F}')\) defined by
\[
\mathcal{F}' = \{ Z \subseteq E : Z = X_1 \cup \ldots \cup X_k \text{ for some } X_1, \ldots, X_k \in \mathcal{F} \}
\]
is an antimatroid. Further, \((E, \mathcal{F})\) is the \(m\)-truncation of \((E, \mathcal{F}')\).

Proof: Consider two sets \(X', Y' \in \mathcal{F}'\). To show that \(\mathcal{F}'\) is an antimatroid it is sufficient by Proposition 1 to demonstrate that \(X' \cup Y' \in \mathcal{F}'\) and that \(\mathcal{F}'\) is a greedoid. Since \(X' \in \mathcal{F}'\), there exist sets \(X_1, \ldots, X_j \in \mathcal{F}\) such that \(X' = X_1 \cup \ldots \cup X_j\). Likewise, there exists sets \(Y_1, \ldots, Y_k \in \mathcal{F}\) such that \(Y' = Y_1 \cup \ldots \cup Y_k\). Certainly, \(X' \cup Y' = X_1 \cup \ldots \cup X_j \cup Y_1 \cup \ldots \cup Y_k\) so that by definition \(X' \cup Y' \in \mathcal{F}'\). To prove that \(\mathcal{F}'\) is a greedoid, assume \(|X'| > |Y'\)| and let \(i\) be an index such that there exists an \(x \in X_i\) but \(x \notin Y'\). As there may be more than one such \(x\), choose some \(X \subseteq X_i\), \(X \in \mathcal{F}\) such that \(|X - Y'| = 1\). This can be done since \(\mathcal{F}\) is a greedoid.

But then \(Y' \cup X \in \mathcal{F}'\) and \(|Y' \cup X| = |Y'| + 1\) proving \(\mathcal{F}'\) is a greedoid.

To show that the \(m\)-truncation of \((E, \mathcal{F}')\) is \((E, \mathcal{F})\) it is sufficient to demonstrate that \(X \in \mathcal{F}\) if and only if \(X \in \mathcal{F}'\) and \(|X| \leq m\). It is clear by the construction of \(\mathcal{F}'\) that \(X \in \mathcal{F}\) implies \(X \in \mathcal{F}'\) and obviously \(|X| \leq m\), completing half of the proof. If \(X \in \mathcal{F}'\) it follows that there exist sets \(X_1, \ldots, X_j \in \mathcal{F}\) such that \(X = X_1 \cup \ldots \cup X_j\). If in addition \(|X| \leq m\), then \(|(X_1 \cup \ldots \cup X_{i-1}) \cup X_i| \leq m\) for \(1 \leq i \leq j\) so that by induction and Proposition 1, \(X = X_1 \cup \ldots \cup X_j \in \mathcal{F}\).

Proposition 2 thus demonstrates that the class of greedoids with the truncated upper interval property is exactly the class of truncated antimatroids, so that henceforth we refer to such greedoids as truncated antimatroids. We state this result in the following proposition for completeness.

Proposition 3: A greedoid \((E, \mathcal{L})\) has the truncated upper interval property if and only if it is a truncated antimatroid.

The following two propositions will be required to prove the main theorem in Section 5.

Proposition 4: Let \((E, \mathcal{L})\) be a truncated antimatroid. If \(\alpha, \beta \in \mathcal{L}\), \(|\alpha| < \rho(\mathcal{L})\) and \(\beta^* - \alpha^* = \{x\}\), then \(ax \in \mathcal{L}\).
Proof: Let \((E, \mathcal{F})\) be the set system associated with \((E, \mathcal{L})\) as defined in Theorem 1. By Proposition 2 it follows that \(\alpha^* \cup \beta^* = \alpha^* \cup \{x\} \in \mathcal{F}\). By Theorem 1, this implies \(\alpha x \in \mathcal{L}\).

Proposition 5: A greedoid \((E, \mathcal{L})\) is a truncated antimatroid if and only if \(\alpha, \alpha \beta, \alpha x \in \mathcal{L}\) with \(x \not\in \beta^*\) and \(|\alpha \beta| < \rho(\mathcal{L}) \Rightarrow \alpha x \beta \in \mathcal{L}\).

Proof: The if half of the proof follows directly from the definition of the truncated upper interval property using a simple inductive argument. The only if half of the proof follows from a simple inductive argument using Proposition 4.

0.5 Results

With all of the preliminary results presented, we are now in a position to complete the main theorem.

Theorem 2: Let \((E, \mathcal{L})\) be a simple language. The greedy algorithm solves the minmax nesting problem for every \(f\)-monotone maximum nesting function \(W\) if and only if \((E, \mathcal{L})\) is a truncated antimatroid.

Proof: (if): Clearly \(e\) is the minimum cost word of length 0. To complete the inductive argument, let \(x_1 \ldots x_k\) be such that for \(i = 1, \ldots, k\), \(W(x_1 \ldots x_k) = \min \{W(\alpha_k : \alpha_k \in \mathcal{L})\}\), and let \(x_{k+1}\) be a greedy choice for \(x_1 \ldots x_k\). If \(x_1 \ldots x_{k+1}\) is not optimal among words of length \(k + 1\), then it follows from the definition of a maximum nesting function that there exists a solution \(y_1 \ldots y_{k+1}\) such that

\[
\max \{f(y_i, (y_1 \ldots y_i)^*)\} \geq \max \{f(x_i, (x_1 \ldots x_i)^*)\} \quad i = 1, \ldots, k
\]

and in addition

\[
\max \{f(y_i, (y_1 \ldots y_i)^*)\} < \max \{f(x_i, (x_1 \ldots x_i)^*)\} \quad i = 1, \ldots, k + 1
\]

Together, these two conditions imply
(a) \( f(y_i, (y_1 \ldots y_i)^*) < f(x_{k+1}, (x_1 \ldots x_{k+1})^*) \) \( i = 1, \ldots, k + 1 \)

Certainly, \((x_1 \ldots x_k)^* \neq (y_1 \ldots y_k)^*\). If so, then since \(y_1 \ldots y_k y_{k+1} \in \mathcal{L}\) and \(x_1 \ldots x_k \in \mathcal{L}\) it follows that \(x_1 \ldots x_k y_{k+1} \in \mathcal{L}\) and clearly \(f(y_{k+1}, (y_1 \ldots y_k y_{k+1})^*) = f(y_{k+1}, (x_1 \ldots x_k y_{k+1})^*)\). However, by (a) this implies \(y_{k+1}\) is a strictly better choice than \(x_{k+1}\), contradicting the fact that \(x_{k+1}\) is a greedy choice.

Thus, let \(j\) be the smallest index such that \(y_j \not\in (x_1 \ldots x_k)^*\). By Proposition 4 letting \(\alpha = x_1 \ldots x_k\) and \(\beta = y_1 \ldots y_j\), it follows that \(x_1 \ldots x_k y_j \in \mathcal{L}\). Further, since \((y_1 \ldots y_j)^* \subseteq (x_1 \ldots x_k y_j)^*\), it follows by the monotonicity of \(f\) that \(f(y_j, (x_1 \ldots x_k y_j)^*) \leq f(y_j, (y_1 \ldots y_j)^*)\). However, again by (a) this implies \(y_j\) is a strictly better choice than \(x_{k+1}\), contradicting the fact that \(x_{k+1}\) is a greedy choice.

(only if): Consider first the case where \((E, \mathcal{L})\) is not accessible. Let \(\alpha \in \mathcal{L}\) be an inaccessible word with \(f(z, A) = 0\) for all \(z \in \alpha^*\), \(A \subseteq E\), and let \(f(z, A) = 1\) for all \(z \not\in \alpha^*\), \(A \subseteq E\). The corresponding maximum nesting function \(W\) defined by \(f\) is trivially \(f\)-monotone. Since \(\alpha\) is inaccessible, the greedy algorithm cannot generate \(\alpha\) and must therefore fail to generate the unique word that minimizes \(W\) for \(\mathcal{L} = |\alpha|\).

Thus, suppose that \((E, \mathcal{L})\) is accessible but is not a greedoid. Then there exist words \(\alpha, \beta \in \mathcal{L}\) with \(|\alpha| < |\beta|\) but \(\alpha x \not\in \mathcal{L}\) for all \(x \in \beta^* - \alpha^* \neq \emptyset\). Define \(f\) as follows.

\[
\begin{align*}
    f(z, A) &= \begin{cases} 
    0 & \text{if } z = \alpha(i) \text{ and } \alpha(q) \in A \text{ for } q = 1, \ldots, i - 1 \\
    |E| & \text{otherwise}
    \end{cases} \\
    f(z, A) &= \begin{cases} 
    1 & \text{if } z = \beta(i), z \not\in \alpha^*, \text{ and } \beta(q) \in A \text{ for } q = 1, \ldots, i - 1 \\
    |E| & \text{otherwise}
    \end{cases} \\
    f(z, A) &= \begin{cases} 
    |E| & \text{for } z \not\in \alpha^* \cup \beta^*
    \end{cases}
\end{align*}
\]

Clearly, \(f\) satisfies \(f(x, A) \leq f(x, B)\) if \(B \subseteq A\). By construction, after \(|\alpha|\) iterations of the greedy algorithm the generated word will be \(\alpha\). If \(\alpha x \not\in \mathcal{L}\) for all \(x \in E - \alpha^*\), then clearly the greedy algorithm has failed to generate a minmax word of cardinality \(|\alpha| + 1 \leq |\beta| \leq \rho(\mathcal{L})\). If there exists an \(x \in E\) such that \(\alpha x \in \mathcal{L}\), then \(x \not\in \beta^*\) by assumption.
Consequently, \( f(x,(\alpha x)^*) = |E| \). On the other hand, \( \beta_{|\alpha|+1} \) is feasible and \( W(\beta_{|\alpha|+1}) = 1 \) so that once again the greedy algorithm fails.

Thus, suppose \((E,\mathcal{L})\) is a greedoid but is not a truncated antimatroid. By Proposition 5, there exist words \( \alpha x, \alpha y_1 \ldots y_k \in \mathcal{L} \) with \( x \not\in (y_1 \ldots y_k)^* \) and \( |\alpha y_1 \ldots y_k| < \rho(\mathcal{L}) \) such that \( \alpha x y_1 \ldots y_k \notin \mathcal{L} \). Let \( j \in \{0,\ldots,k-1\} \) be the smallest index such that \( \alpha x y_1 \ldots y_{j+1} \notin \mathcal{L} \), and define \( f \) as follows.

\[
\begin{align*}
  f(z, A) & = \begin{cases} 
  0 & \text{if } z = \alpha(i) \text{ and } \alpha(q) \in A \text{ for } q = 1, \ldots, i-1 \\
  |E| & \text{otherwise}
  \end{cases} \\
  f(x, A) & = \begin{cases} 
  0 & \text{if } \alpha^* \subseteq A \\
  |E| & \text{otherwise}
  \end{cases} \\
  f(y_i, A) & = \begin{cases} 
  1 & \text{if } y_q \in A \text{ for } q = 1, \ldots, i-1 \\
  |E| & \text{otherwise}
  \end{cases} \\
  f(z, A) & = |E| \text{ for } z \not\in \alpha^* \cup \{x\} \cup \{y_1, \ldots, y_k\}
\end{align*}
\]

Clearly, \( f(x, A) \leq f(x, B) \) if \( B \subseteq A \). After \(|\alpha| + 1 + j \) iterations of the greedy algorithm the generated word will be \( \alpha x y_1 \ldots y_j \), and since \( |\alpha x y_1 \ldots y_j| \leq |\alpha y_1 \ldots y_k| < \rho(\mathcal{L}) \), there exists some feasible choice for \( \alpha x y_1 \ldots y_j \). By construction, \( y_{j+1} \) is not a feasible choice. Consequently, all feasible choices \( z \) have \( f(z,(\alpha x y_1 \ldots y_j z)^*) = |E| \). On the other hand, \( \alpha y_1 \ldots y_{j+1} \) is feasible and \( W(\alpha y_1 \ldots y_{j+1}) \leq 1 \).

### 0.6 Examples

Theorem 2 not only provides an algorithmic characterization of antimatroids but extends Lawler's result to this more general class of combinatorial structures. The following are a small set of examples of problems captured by Theorem 2.

#### 0.6.1 Job Scheduling in a Deflationary Period

Consider the single machine job scheduling problem mentioned earlier in the following context. A company is faced with a set of jobs \( \mathcal{E} \) that must
be completed subject to a set of precedence constraints. Only one job can be scheduled at a time, but the company wishes to complete the jobs by the earliest possible time so that whatever the order chosen for the jobs, when one job is completed the next will commence. Each job \( x \) has a fixed completion time \( p_x \), and the work is being scheduled in a deflationary period so that the later a job is commenced the less it will cost. The problem is to find a schedule satisfying the precedence constraints that minimizes the maximum of the incurred job costs.

This is an example of a minmax nesting problem on a class of antimatroids, namely poset greedoids, and so by Theorem 2 it is solvable by the greedy algorithm. The poset is defined by the scheduling constraints, and the underlying \( f \) is defined by

\[
f(x, A) = c_x \left( \sum_{y \in A} p_y \right)
\]

where \( c_x(t) \) is assumed to be nonincreasing and \( t = 0 \) is the time the first job commences. Clearly, \( f \) satisfies \( f(x, A) \leq f(x, B) \) if \( B \subseteq A \).

As an alternative to the deflationary environment assumption, it is possible to consider a period of constant costs but with discounting included. Under this assumption it is clear that the discounted job costs are nonincreasing. It is easy to see that in this instance the greedy algorithm dictates that the least expensive job available should always be chosen.

It is valuable to note that in order to apply the greedy algorithm the \( c_x(t) \) do not need to be known in advance. When a job is finished at time \( \bar{t} \), \( c_x(\bar{t}) \) can be calculated for all jobs that can commence at time \( \bar{t} \) and the job with the minimum cost chosen. Having such a weak requirement on knowledge of the \( c_x(t) \) is extremely important since it implies that a company only needs to have a qualitative rather than quantitative projection of costs. It is also valuable to note that the proof of Theorem 2 demonstrates that any set of \( k \) jobs chosen by the greedy algorithm minimizes the maximum job cost over all choices and orderings of \( k \) jobs.

The problem can also be stated in terms of a company seeking to maximize the minimum profit among a set of contracts in an inflationary period. An interesting specific example might be that of a building contractor choosing among home construction projects in a housing market where prices are generally rising. The name greedy algorithm also seems particularly relevant
in this context.

0.6.2 Road Construction in a Deflationary Period

Consider a construction company charged with the task of constructing a road network \( E \) connecting a set of locations \( V \). The construction equipment is initially located at location \( r \in V \). Since the equipment needs a road on which to travel when it is relocated, construction cannot begin on the road connecting locations \( x \) and \( y \) until roads have been constructed linking \( r \) to \( x \) or \( y \).

Only one road can be constructed at a time, but the construction company must complete the roads at the earliest possible time so that whatever the order chosen for construction, when one road is completed construction on the next road commences immediately. Each road \( x \) has a fixed completion time \( p_x \), and the work is being scheduled in a deflationary period so that the later construction of a road is commenced the less it will cost. The problem is to find a feasible construction schedule that minimizes the maximum of the incurred road construction costs.

This is an example of a minmax nesting problem on a class of antimatroids called \textit{undirected dense branching greedoids} [Boyd 1987], and so by Theorem 2 it is solvable by the greedy algorithm. The underlying \( f \) is defined by

\[
f(x, A) = c_x \left( \sum_{y \in A} p_y \right)
\]

where \( c_x(t) \) is assumed to be nonincreasing and \( t = 0 \) is the time construction on the first road commences. It is clearly the case that \( f \) satisfies \( f(x, A) \leq f(x, B) \) if \( B \subseteq A \).

0.6.3 Orderly Retreat

Consider an army in the midst of battle with a set of fixed emplacements \( E \). The objective is to perform an orderly retreat that minimizes the maximum loss of equipment incurred at any emplacement. Assume that a longer preparation time implies a better prepared retreat for an emplacement and thus a lower cost. So that no emplacement is left unprotected, one emplacement
retreats at a time, and the retreating emplacement is always restricted to an emplacement on the front.

If the front is interpreted as the extreme points of the convex polytope defined by the remaining emplacements, this is an example of a minmax nesting function on a class of antimatroids called *convex shelling greedoids* [Korte and Lovász 1982] and so by Theorem 2 it is solvable by the greedy algorithm. If $p_x$ is the time required for emplacement $x$ to retreat, then the underlying $f$ is defined by

$$f(x, A) = c_x \left( \sum_{y \in A} p_y \right)$$

where $c_x(t)$ is assumed to be nonincreasing and $t = 0$ is the time the retreat is initiated. Clearly, $f$ satisfies $f(x, A) \leq f(x, B)$ if $B \subseteq A$.

### 0.7 Acknowledgements

The author would like to thank Joe Ferreira, Tom Magnanti, Jim Orlin, and David Shmoys for their comments on an earlier version of this material.
References


