# Random order greedy up to 4 parts*

Yuval Filmus

November 14, 2018

## 1 Introduction

The greedy algorithm is a heuristic for optimizing a monotone set function $f$ given a constraint $\mathcal{I} \subseteq 2^{\mathcal{U}}$, in other words for solving the following optimization problem:

$$\max_{S \in \mathcal{I}} f(S).$$

Assuming that $\mathcal{I}$ is hereditary (that is, if $A \in \mathcal{I}$ then all subsets of $A$ are also in $\mathcal{I}$), the greedy algorithm can be stated as follows:

1. Initialize $S \leftarrow \emptyset$.
2. While there exists an element $x \in \mathcal{I} \setminus S$ such that $S \cup \{x\} \in \mathcal{I}$:
$$S \leftarrow S \cup \{x\}, \text{ where } x = \operatorname*{arg\,max}_{y\colon S \cup \{y\} \in \mathcal{I}} f(S \cup \{y\}).$$
3. Output $S$.

The performance of this heuristic naturally depends on the nature of both $f$ and $\mathcal{I}$. The simplest case is when $f$ is a monotone *linear function*, which is a function that satisfies $f(\emptyset) = 0$ and $f(S \cup T) = f(S) + f(T)$ whenever $S, T$ are disjoint; equivalently, $f(S) = \sum_{x \in S} w(x)$, where $w\colon \mathcal{U} \to \mathbb{R}_{\geq 0}$ is an arbitrary non-negative weight function. In this case, it is classically known that the greedy heuristic is optimal — produces the optimal value for all objective functions $f$ — if and only if $\mathcal{I}$ is a *matroid* [Rad57, Edm71].

The picture becomes more interesting when $f$ is a monotone submodular function, that is, when it satisfies the following three properties:

(a) Normalization: $f(\emptyset) = 0$.

(b) Monotonicity: if $A \subseteq B$ then $f(A) \leq f(B)$.

(c) Submodularity: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.
Equivalently, if $A \subseteq B$ then $f(C|A) \geq f(C|B)$, where $f(C|A) := f(C \cup A) - f(A)$.

*Based on work dating 2010–2011.

The standard example of a monotone submodular function is a *coverage function*: the elements of $\mathcal{U}$ are interpreted as subsets of a universe $\mathcal{V}$ with a non-negative weight function $w\colon \mathcal{V} \to \mathbb{R}_{\geq 0}$, and $f(S)$ is the total weight of all elements (of $\mathcal{V}$) covered by the sets in $S$.

When $\mathcal{I}$ consists of all subsets of $\mathcal{U}$ of size at most $r$ — the so-called *uniform matroid* — the greedy algorithm has an approximation ratio of $1-1/e$ (or even slightly better, for fixed $r$) [NWF78]. For general matroids, however, the approximation ratio drops to $1/2$ [FNW78]. A simple example showing the tightness of $1/2$ is given by a coverage function over a *partition matroid*. In a partition matroid, the universe $\mathcal{U}$ is partitioned into $r$ sets $P_1, \ldots, P_r$, and the set $\mathcal{I}$ consists of all subsets of $P_1 \times \cdots \times P_r$. Here is the example:

$$P_1 = \{A = \{x\}, B = \{y, z\}\}, \qquad P_2 = \{C = \{y\}\}, \qquad w(x) = w(y) = 1, w(z) = \epsilon > 0.$$

The optimal solution is $\{A, C\}$, of value 2, but the greedy algorithm produces the solution $\{B, C\}$, whose value is only $1 + \epsilon$. In this example, the element $z$ guarantees that the algorithm chooses $B$ in the first step. It is more convenient to allow the algorithm to break ties adversarially, in which case we can dispense with this extra element.

There is a simple modification of the greedy algorithm, for the specific case in which $\mathcal{I}$ is a partition matroid with parts $P_1, \ldots, P_r$, which overcomes this counterexample. Given a permutation $\pi$ of $\{1, \ldots, r\}$, the algorithm proceeds as follows:

1. Initialize $S \leftarrow \emptyset$.
2. For $i = \pi(1), \ldots, \pi(r)$:

$$S \leftarrow S \cup \{x\}, \text{ where } x = \arg\max_{y \in P_i} f(S \cup \{y\}).$$

3. Return $S$.

If we choose the permutation $\pi = 12$, then the greedy algorithm still fails on the aforementioned example. But when $\pi = 21$, it finds the optimal solution! This suggests running the modified greedy algorithm on all possible orderings. However, this is not feasible unless $r$ is very small, since there are $r!$ many different orderings. Instead, it is natural to consider picking $\pi$ *at random*. We call this algorithm *random order greedy*. We are interested in analyzing the *expected* approximation ratio of this algorithm, as a function of $r$.

The algorithm, as stated, is not completely defined, since we haven't specified a tie-breaking rule. Our analysis will be oblivious to the tie-breaking rule used, as long as the following natural property holds:

*The tie-breaking rule used to select $x \in P_{\pi(j)}$ is independent of the relative order of $\pi(j+1), \ldots, \pi(n)$.*

This assumption is especially natural if we consider random order greedy as an online algorithm, in which the parts $P_1, \ldots, P_r$ arrive in random order and are revealed to the algorithm one at a time. Whenever each part is revealed, the algorithm must immediately choose some element from the part to add to the solution.

**Definition 1.1.** Given $r \geq 1$, we define $\rho(r)$ as the supremum value $\rho^*$ such that the expected approximation ratio of random order greedy is at least $\rho^*$ on *any* instance.

To illustrate the definition, the example above shows that $\rho(r) \leq (1/2 + 1)/2 = 3/4$.

It is always the case that $\rho(r) \geq 1/2$. Recently, Buchbinder, Feldman and Garg [BFG] have shown that $\rho(r) \geq c$ for some absolute constant $c > 1/2$ independent of $r$. In this short note, we prove the following result:

**Theorem 1.2.** *We have* $\rho(1) = 1$, $\rho(2) = 2/3$, $\rho(3) = 7/12$, *and* $103/180 \leq \rho(4) \leq 19/33$.

Since $7/12 < 1-1/e$, this shows that random order greedy performs worse than other approximation algorithms such as continuous greedy [CCPV11, FNS11] or non-oblivious local search [FW14], which give the optimal approximation ratio $1 - 1/e$ for *any* matroid.

## 2 Two parts

In this section we prove that $\rho(2) = 2/3$. The proof consists of two parts: an upper bound and a lower bound. The upper bound is given by a simple construction, based on a coverage function. All other upper bounds in this note are also based on coverage functions, which we conjecture are always enough to obtain the upper bound on $\rho(r)$ for any $r$.

### 2.1 Upper bound

The upper bound stems from the following set system:

| $P_1$ | $P_2$ |
|---|---|
| $O_1 = \{a_1, b_1, c_1\}$ | $O_2 = \{a_2, b_2, c_2\}$ |
| $S_1 = \{a_2, b_2, c_1\}$ | $S_2 = \{a_2, b_1, c_1\}$ |

All elements have weight 1 in this case.

It is easy to check that $\{O_1, O_2\}$ is the optimal solution, which contains 6 elements. In contrast, the random order greedy could operate as follows (breaking ties adversarially):

- When $\pi = 12$: Choose $S_1$ in the first step. Any choice in the second step will result in a solution containing only 4 elements.
- When $\pi = 21$: Choose $S_2$ in the first step. Any choice in the second step will result in a solution containing only 4 elements.

This shows that the approximation ratio of random order greedy on this example is $4/6 = 2/3$.

As mentioned above, by adding small-weight elements to $S_1, S_2$ we can force random order greedy to choose $S_1, S_2$ (even non-adversarially), at an arbitrarily small cost in the resulting approximation ratio.

### 2.2 Lower bound

For the lower bound, we use the following notation:

- $o_1, o_2$ is an optimal solution.
- $s_1, s_{12}$ is the solution output by the algorithm when $\pi = 12$; note $s_1 \in P_1$ and $s_{12} \in P_2$.
- $s_2, s_{21}$ is the solution output by the algorithm when $\pi = 21$.

Since the algorithm always makes a choice which maximizes the value of the objective function, we know that the following inequalities must hold, where we use the shorthand notation $f(x_1, \dots, x_\ell) = f(\{x_1, \dots, x_\ell\})$:

- Considering the first step when $\pi = 12$: $f(s_1) \geq f(o_1), f(s_{21})$.
- Considering the first step when $\pi = 21$: $f(s_2) \geq f(o_2), f(s_{12})$.
- Considering the second step when $\pi = 12$: $f(s_{12}|s_1) \geq f(o_2|s_1), f(s_2|s_1)$.
- Considering the second step when $\pi = 21$: $f(s_{21}|s_2) \geq f(o_1|s_2), f(s_1|s_2)$.

From these inequalities, we can deduce

$$\begin{aligned} 3f(s_1, s_{12}) &= 3f(s_1) + 3f(s_{12}|s_1) \\ &\geq f(s_1) + 2f(o_1) + 2f(o_2|s_1) + f(s_2|s_1) \\ &= 2f(o_1) + 2f(o_2|s_1) + f(s_1, s_2). \end{aligned}$$

Switching the roles of 1 and 2, we similarly deduce

$$3f(s_2, s_{21}) \geq 2f(o_2) + 2f(o_1|s_2) + f(s_1, s_2).$$

Summing both inequalities gives

$$3f(s_1, s_{12}) + 3f(s_2, s_{21}) \geq 2f(o_1) + 2f(o_2) + 2f(o_1|s_2) + 2f(o_2|s_1) + 2f(s_1, s_2).$$

We now turn our gaze to the right-hand side: clearly $f(o_1) + f(o_2) \geq f(o_1, o_2)$, and moreover

$$\begin{aligned} f(o_1|s_2) + f(o_2|s_1) + f(s_1, s_2) \geq f(o_1|o_2, s_1, s_2) + f(o_2|s_1, s_2) + f(s_1, s_2) \geq \\ f(o_1|o_2, s_1, s_2) + f(o_2, s_1, s_2) \geq f(o_1, o_2, s_1, s_2) \geq f(o_1, o_2). \end{aligned}$$

In total, this shows that

$$3f(s_1, s_{12}) + 3f(s_2, s_{21}) \geq 4f(o_1, o_2),$$

or equivalently,

$$\frac{f(s_1, s_{12}) + f(s_2, s_{21})}{2} \geq \frac{2}{3} f(o_1, o_2).$$

The left-hand side is the expected value of the solution produced by the greedy algorithm, and so this completes the proof that the approximation ratio is always at least 2/3.

## 3 A lower bound method

The main inequality driving the lower bound on $\rho(2)$ is

$$f(o_1|s_2) + f(o_2|s_1) + f(s_1, s_2) \geq f(o_1, o_2).$$

In this section we discuss a general method of proving such inequalities, using a kind of proof system constructed expressly for this purpose.

**Definition 3.1.** Let $\mathcal{U}$ be a universe. A *line* is an expression of the form $\alpha \to b$, where $\alpha \subseteq \mathcal{U}$ and $b \in \mathcal{U}$. A *proof* is a sequence of lines $\alpha_i \to b_i$ such that $\alpha_i \subseteq \{b_j : j < i\}$.

As an example, here is a proof:

$$\begin{aligned} &\to s_1 \\ s_1 &\to s_2 \\ s_1 &\to o_2 \\ s_2 &\to o_1 \end{aligned}$$

We will abbreviate lines with the same left-hand side by combining their right-hand sides. For example, the proof above can be shortened to

$$\begin{aligned} &\to s_1 \\ s_1 &\to s_2, o_2 \\ s_2 &\to o_1 \end{aligned}$$

The main result about proofs is the following lemma, which generalizes the aforementioned main inequality.

**Lemma 3.2.** *Let $(\alpha_i \to b_i)_{1 \le i \le \ell}$ be a proof. Then every monotone submodular function $f$ on $\mathcal{U}$ satisfies the following inequality:*

$$\sum_{i=1}^{\ell} f(b_i|\alpha_i) \ge f(b_1, \dots, b_\ell).$$

*Proof.* The proof is by induction on $\ell$. When $\ell = 1$, necessarily $\alpha_1 = \emptyset$, and so the inequality is the tautological $f(b_1) \ge f(b_1)$.

Now suppose that the claim is true for $\ell$, and consider an additional line $\alpha_{\ell+1} \to b_{\ell+1}$, where $\alpha_{\ell+1} \subseteq \{b_1, \dots, b_\ell\}$. Using the induction hypothesis,

$$\sum_{i=1}^{\ell+1} f(b_i|\alpha_i) \ge f(b_1, \dots, b_\ell) + f(b_{\ell+1}|\alpha_{\ell+1}) \ge f(b_1, \dots, b_\ell) + f(b_{\ell+1}|\beta_1, \dots, b_\ell) = f(b_1, \dots, b_{\ell+1}). \quad \square$$

The next step is relating this lemma to random order greedy, as applied to a partition matroid with $r$ parts $P_1, \dots, P_r$. We first need to fix some notations. We denote an optimal solution by

$$o_1, \dots, o_r.$$

When run on a permutation $\pi$ of $\{1, \dots, r\}$, random order greedy selects elements from the parts $P_{\pi(1)}, \dots, P_{\pi(r)}$, in this order. Furthermore, the element chosen from $P_{\pi(j)}$ depends only on the ones chosen in the preceding steps, as a consequence of our assumption on the tie-breaking rule. Therefore we can denote the sets chosen by random order greedy on this permutation by

$$s_{\pi(1)}, s_{\pi(1)\pi(2)}, \dots, s_{\pi(1)\dots\pi(r)}.$$

We think of the elements as *colored* by the parts they belong to: $o_i$ has color $i$ and $s_{\pi(1)\dots\pi(i)}$ has color $\pi(i)$.

We can now describe the specific type of proofs which will be useful for analyzing random order greedy.

**Definition 3.3.** A line $\alpha \to b$ is *legal* if $\alpha = s_{\pi(1)}, s_{\pi(1)\pi(2)}, \ldots, s_{\pi(1)\ldots\pi(t)}$ for some permutation $\pi$, and $b$ has color $\pi(t+1)$.

A proof is *legal* if all lines are legal, and furthermore each of $o_1, \ldots, o_r$ appears on the right-hand side of some line.

A legal proof corresponds to an upper bound on $f(o_1, \ldots, o_r)$, as indicated by the following simple lemma.

**Lemma 3.4.** *Let $(\alpha_i \to b_i)_{1 \le i \le \ell}$ be a legal proof. Suppose $\alpha_i \to b_i$ corresponds to a permutation $\pi_i$, where $\alpha_i$ has length $t_i$. Then for every monotone submodular function $f$, the outcome of running random order greedy satisfies*

$$\sum_{i=1}^{\ell} f(s_{\pi_i(1)\ldots\pi_i(t_i+1)} | s_{\pi_i(1)}, \ldots, s_{\pi_i(1)\ldots\pi_i(t_i)}) \ge f(o_1, \ldots, o_r).$$

*Proof.* The semantics of random order greedy guarantees that for every line $i$,

$$f(s_{\pi_i(1)\ldots\pi_i(t_i+1)} | s_{\pi_i(1)}, \ldots, s_{\pi_i(1)\ldots\pi_i(t_i)}) \ge f(b_i | s_{\pi_i(1)}, \ldots, s_{\pi_i(1)\ldots\pi_i(t_i)}) = f(b_i | \alpha_i).$$

Therefore Lemma 3.2 implies that

$$\sum_{i=1}^{\ell} f(s_{\pi_i(1)\ldots\pi_i(t_i+1)} | s_{\pi_i(1)}, \ldots, s_{\pi_i(1)\ldots\pi_i(t_i)}) \ge \sum_{i=1}^{\ell} f(b_i | \alpha_i) \ge f(b_1, \ldots, b_\ell) \ge f(o_1, \ldots, o_r),$$

using the assumption that all of $o_1, \ldots, o_r$ appear in $b_1, \ldots, b_\ell$. □

The next step is *symmetrization.* The inequality in the statement of Lemma 3.4 remains valid if we apply a random permutation of the parts. Such a random permutation turns each quantity of the form $f(s_{1\ldots t+1} | s_1, \ldots, s_{1\ldots t})$ into the symmetrized quantity

$$S_t := \mathop{\mathbb{E}}_{\pi}[f(s_{\pi(1)\ldots\pi(t+1)} | s_{\pi(1)}, \ldots, s_{\pi(1)\ldots\pi(t)})].$$

This allows us to deduce a clean corollary of Lemma 3.4, which requires first a definition.

**Definition 3.5.** The *weight* of a legal proof $(\alpha_i \to b_i)_{1 \le i \le \ell}$ is the vector $w_0, \ldots, w_{r-1}$, where $w_t$ is the number of lines in which $|\alpha_i| = t$.

For example, the legal proof considered above has weight $(1, 3)$. Another legal proof, of weight $(2, 0)$, is the trivial

$$\to o_1, o_2.$$

**Lemma 3.6.** *If a legal proof has weight $w$ then*

$$\sum_{t=0}^{r-1} w_t S_t \ge f(o_1, \ldots, o_r).$$

*Proof.* This follows straightforwardly from Lemma 3.4 by symmetrization, as indicated above. □

Let us denote by $S$ the expected value of the solution produced by random order greedy:

$$S = \mathop{\mathbb{E}}_{\pi}[f(s_{\pi(1)}, \ldots, s_{\pi(1)\ldots\pi(r)})].$$

We can relate $S$ to $S_0, \ldots, S_{r-1}$ as follows:

**Lemma 3.7.** *We have $S = S_0 + \cdots + S_{r-1}$.*

*Proof.* Simple application of the chain rule $f(a, b) = f(a) + f(b|a)$. $\square$

We are now ready to state the main result of this section:

**Theorem 3.8.** *Suppose $w^1, \ldots, w^s$ are weight vectors of legal proofs, and let $\alpha_1, \ldots, \alpha_s \geq 0$ satisfy the constraint*

$$\alpha_1 w^1 + \cdots + \alpha_s w^s = (1, \ldots, 1).$$

*Then the approximation ratio of random order greedy on $r$ parts satisfies*

$$\rho(r) \geq \alpha_1 + \cdots + \alpha_s.$$

*Proof.* Combining Lemma 3.7 and Lemma 3.6, we obtain

$$S \geq S_1 + \cdots + S_r = \sum_{k=1}^{s} \alpha_k \sum_{t=0}^{r-1} w_t^k S_t \geq \sum_{k=1}^{s} \alpha_k f(o_1, \ldots, o_r). \qquad \square$$

As an illustration, above we have shown that $w^1 = (1, 3)$ and $w^2 = (2, 0)$ are weight vectors of legal proofs. Taking $\alpha_1, \alpha_2 = 1/3, 1/3$, we see that $\rho(2) \geq 2/3$.

# 4 Three parts

In this section we prove that $\rho(3) = 7/12$, leveraging our work in Section 3.

## 4.1 Upper bound

The upper bound stems from the following set system:

| $P_1$ | $P_2$ | $P_3$ |
|---|---|---|
| $O_1 = \{x_1, y_1, z_1, w_1\}$ | $O_2 = \{x_2, y_2, z_2, w_2\}$ | $O_3 = \{x_3, y_3, z_3, w_3\}$ |
| $S_1 = \{x_2, y_2, x_3, y_3\}$ | $S_2 = \{x_1, y_1, x_3, y_3\}$ | $S_3 = \{x_1, y_1, x_2, y_2\}$ |
| | $S_{12} = \{x_1, x_2, z_3\}$ | $S_{13} = \{x_1, x_3, z_2\}$ |
| $S_{21} = \{x_1, x_2, z_3\}$ | | $S_{23} = \{x_2, x_3, z_1\}$ |
| $S_{31} = \{x_1, x_3, z_2\}$ | $S_{32} = \{x_2, x_3, z_1\}$ | |

As in Section 2.1, all elements have weight 1. Note that $S_{ij} = S_{ji}$.

It is easy to see that $\{O_1, O_2, O_3\}$ is the optimal solution. By choosing an appropriate tie-breaking rule (or by suitably adding elements of small weight), we can guarantee that when the permutation is $ijk$, random order greedy chooses the sets $S_i, S_{ij}, S_{ijk}$. To see this, it suffices to check the case $ijk = 123$, by symmetry. We need to check the following conditions:

$$|S_1| \geq |O_1|, |S_{21}|, |S_{31}|,$$
$$|S_1 \cup S_{12}| \geq |S_1 \cup O_2|, |S_1 \cup S_2|, |S_1 \cup S_{32}|,$$
$$|S_1 \cup S_{12} \cup O_3| \geq |S_1 \cup S_{12} \cup S_3|, |S_1 \cup S_{12} \cup S_{13}|, |S_1 \cup S_{12} \cup S_{23}|.$$

Substituting the actual values, we get

$$4 \geq 4, 3, 3, \quad 6 \geq 6, 6, 5, \quad 7 \geq 7, 7, 7.$$

The solution produced by random order greedy has value 7, compared to the optimal 12.

#### 4.1.1 Symmetric upper bounds

The upper bound proved in this section is *symmetric*: it is invariant under permutations of the three indices $1, 2, 3$ (curiously, the upper bound in Section 2.1 doesn't satisfy this property). While symmetric upper bounds are not the most general ones, they are easier to construct and analyze. In particular, it suffices to describe the following sets:

$$O_1, S_1, S_{12}, \ldots, S_{1\ldots r}.$$

In all our examples, in fact $S_{1\ldots r} = O_r$.

The upper bound can thus be described succinctly as follows:

$$O_1 = \{x_1, y_1, z_1, w_1\}$$
$$S_1 = \{x_2, y_2, x_3, y_3\}$$
$$S_{12} = \{x_1, x_2, z_3\}$$

### 4.2 Lower bound

For the lower bound, we use the method of Section 3. We start by exhibiting three legal proofs.

The first legal proof has the trivial weight vector $(3, 0, 0)$:

$$\to o_1, o_2, o_3$$

The second legal proof has weight vector $(1, 3, 2)$:

$$\begin{aligned} &\to s_2 \\ s_2 &\to o_3, s_{23} \\ s_2, s_{23} &\to o_1, s_1 \\ s_1 &\to o_2 \end{aligned}$$

The third legal proof has weight vector $(1, 2, 4)$:

$$\begin{aligned} &\to s_1 \\ s_1 &\to s_{12} \\ s_1, s_{12} &\to o_3, s_{13} \\ s_1, s_{13} &\to o_2, s_2 \\ s_2 &\to o_1 \end{aligned}$$

The following equation shows, via Theorem 3.8, that $\rho(3) \geq 14/24 = 7/12$:

$$5(3,0,0) + 6(1,3,2) + 3(1,2,4) = 24(1,1,1).$$

# 5 Four parts

In this section we prove that $103/180 \leq \rho(4) \leq 19/33$.

## 5.1 Upper bound

The upper bound is given by a set system, which we describe along the lines of Section 4.1.1:

$$\begin{aligned}
O_1 &= \{a_1, b_1, c_1, d_1, e_1, f_1, g_1\} \\
S_1 &= \{b_2, c_2, b_3, c_3, b_4, c_4\} \\
S_{12} &= \{c_1, e_2, d_3, e_3, f_3, d_4, e_4, f_4\} \\
S_{123} &= \{f_1, f_2, g_4\}
\end{aligned}$$

This set system has the additional symmetry that $S_{ijk} = S_{jik}$.

In contrast to all examples described above, this time different elements have different weights:

| $x$ | $a_i$ | $b_i$ | $c_i$ | $d_i$ | $e_i$ | $f_i$ | $g_i$ |
|---|---|---|---|---|---|---|---|
| $w(x)$ | 14 | 14 | 8 | 5 | 4 | 7 | 14 |

The optimal solution is $\{O_1, O_2, O_3, O_4\}$. A long but routine calculation shows that random order greedy could choose the sets $S_1, S_{12}, S_{123}, O_4$ when the permutation is 1234. Since $f(O_1, O_2, O_3, O_4) = 264$ while $f(S_1, S_{12}, S_{123}, O_4) = 152$, this shows that $\rho(4) \leq 152/264 = 19/33$.

We mention without proof that a more complicated example (which isn't symmetric) gives an improved upper bound of 207/361.

## 5.2 Lower bound

The lower bound follows via the method of Section 3. This time we need four different legal proofs.

The first legal proof has the trivial weight vector $(4,0,0,0)$:

$$\to o_1, o_2, o_3, o_4$$

The second legal proof has weight vector $(2,3,0,0)$:

$$\begin{aligned}
&\to s_1, o_1 \\
s_1 &\to o_2, o_3, o_4
\end{aligned}$$

The third legal proof has weight vector $(1,1,6,0)$:

$$\begin{aligned}
&\to s_1 \\
s_1 &\to s_{12} \\
s_1, s_{12} &\to s_3, s_{34}, o_3, o_4 \\
s_3, s_{34} &\to o_1, o_2
\end{aligned}$$

The final legal proof has weight vector $(1, 1, 3, 5)$:

$$\begin{aligned} &\to s_1 \\ s_1 &\to s_{12} \\ s_1, s_{12} &\to s_{123} \\ s_1, s_{12}, s_{123} &\to s_{124}, s_{34}, o_4 \\ s_1, s_{12}, s_{124} &\to s_3, o_3 \\ s_3, s_{34} &\to o_1, o_2 \end{aligned}$$

The following equation shows, via Theorem 3.8, that $\rho(4) \geq 103/180$:

$$11(4, 0, 0, 0) + 44(2, 3, 0, 0) + 12(1, 1, 6, 0) + 36(1, 1, 3, 5) = 180(1, 1, 1, 1).$$

# 6 Methodology

In this final section, we explain briefly how we used computer searches to find the proofs in the preceding sections.

## 6.1 Upper bounds

We found all upper bounds using the framework of Section 4.1.1. Let us take as an example the upper bound in Section 4.1. The most general form of a symmetric upper bound is

$$\begin{aligned} &O_1 = A_1 \\ &S_1 = B_1 \cup C_2 \cup C_3 \\ &S_{12} = D_1 \cup E_2 \cup F_3 \end{aligned}$$

We will construct a linear program whose value is the best upper bound that can be obtained in this way. The variables of the linear program are the weights of the elements in the Venn diagram of $A, B, C, D, E, F$ (64 in total). Section 4.1 states 9 linear inequalities that these non-negative weights must satisfy in order to satisfy the semantics of random order greedy for the permutation 123. We specify that the total weight of $O_1, O_2, O_3$ is 1, and minimize the total weight of $S_1, S_{12}, O_3$. The value of this program is 7/12, and one optimal solution is given by

$$\begin{aligned} &w(A \cap \bar{B} \cap \bar{C} \cap \bar{D} \cap \bar{E} \cap \bar{F}) = w(A \cap \bar{B} \cap C \cap \bar{D} \cap \bar{E} \cap \bar{F}) = \\ &w(A \cap \bar{B} \cap C \cap D \cap E \cap \bar{F}) = w(A \cap \bar{B} \cap \bar{C} \cap \bar{D} \cap \bar{E} \cap F) = \frac{1}{12}. \end{aligned}$$

This is exactly the solution appearing in Section 4.1 (scaled by a factor of 12), in which the corresponding elements are called $w, y, x, z$.

## 6.2 Lower bounds

We found all lower bounds using the framework of Section 3. The idea is to enumerate all "minimal" legal proofs, compute their weight vectors $w^1, \ldots, w^s$, and then solve a linear program which maximizes $\alpha_1 + \cdots + \alpha_s$ (over non-negative $\alpha_1, \ldots, \alpha_s$) subject to the constraint $\alpha_1 w^1 + \cdots + \alpha_s w^s = (1, \ldots, 1)$.

To enumerate all legal proofs, we repeatedly come up with elements $b$ such that $\alpha \to b$ can be added to the proof. Since we are only interested in the resulting weight vector, we want to know what are the possible sizes of $\alpha$. If $\alpha \to b$ then in fact $\alpha' \to b$ for all prefixes $\alpha'$ of $\alpha$. Therefore, the possible sizes of $\alpha$ are always $0, \ldots, t$ for some $t$. As an optimization, we only consider the maximal choice $t$. If the resulting weight vector is $(w_0, \ldots, w_{r-1})$, then other weight vectors can be obtained by "shifting mass to the left" (an operataion corresponding to majorization).

Instead of explicitly generating all weight vectors from the "maximal" weight vectors considered above, we change the condition $w := \alpha_1 w^1 + \cdots + \alpha_s w^s = (1, \ldots, 1)$ to the condition "$(1, \ldots, 1)$ can be obtained from $w$ by shifting mass to the left", or equivalently, $w_0 + \cdots + w_t \leq t$ for $0 \leq t \leq r-1$. (The condition $w_0 + \cdots + w_{r-1} = r$ follows from our maximizing $\alpha_1 + \cdots + \alpha_s$.)

## References


[BFG] Niv Buchbinder, Moran Feldman, and Mohit Garg. Online submodular maximization: Beating 1/2 made simple. Manuscript.

[CCPV11] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[Edm71] Jack Edmonds. Matroids and the greedy algorithm. *Math. Programming*, 1(1):127–136, 1971.

[FNS11] Feldman, Joseph (Seffi) Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Proc. IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, 2011.

[FNW78] Marshall L. Fisher, George L. Nemhauser, and Leonard A. Wolsey. An analysis of approximations for maximizing submodular set functions—II. *Math. Programming*, 8:73–87, 1978.

[FW14] Yuval Filmus and Justin Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. *SIAM J. Comp.*, 43:514–542, 2014.

[NWF78] George L. Nemhauser, Leonard A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Math. Programming*, 1:265–294, 1978.

[Rad57] Richard Rado. Note on independence functions. *Proc. of the London Math. Soc.*, s3-7(1):300–320, 1957.