Maximizing Submodular Functions under Matroid Constraints by Evolutionary Algorithms

Tobias Friedrich friedrich@hpi.de
Hasso Plattner Institute, Potsdam, Germany

Frank Neumann frank.neumann@adelaide.edu.au
School of Computer Science, University of Adelaide, Australia

doi:10.1162/EVCO_a_00159

Abstract

Many combinatorial optimization problems have underlying goal functions that are submodular. The classical goal is to find a good solution for a given submodular function *f* under a given set of constraints. In this paper, we investigate the runtime of a simple single objective evolutionary algorithm called (1 + 1) EA and a multiobjective evolutionary algorithm called GSEMO until they have obtained a good approximation for submodular functions. For the case of monotone submodular functions and uniform cardinality constraints, we show that the GSEMO achieves a $(1 - 1/e)$-approximation in expected polynomial time. For the case of monotone functions where the constraints are given by the intersection of $k \geq 2$ matroids, we show that the (1 + 1) EA achieves a $(1/k + \delta)$-approximation in expected polynomial time for any constant $\delta > 0$. Turning to nonmonotone symmetric submodular functions with $k \geq 1$ matroid intersection constraints, we show that the GSEMO achieves a $1/((k + 2)(1 + \varepsilon))$-approximation in expected time $\mathcal{O}(n^{k+6} \log(n)/\varepsilon)$.

Keywords

Submodular functions, matroid constraints, approximation, multiobjective optimization, hypervolume indicator, maximum cut, runtime, theory.

1 Introduction

Evolutionary algorithms can efficiently find the minima of convex functions. While this is known and well studied in the continuous domain, it is not obvious what an equivalent statement for discrete optimization looks like. Let us recall that a differentiable fitness function $f : \mathbb{R} \to \mathbb{R}$ is called *convex* if its derivative $\frac{d}{dx} f(x)$ is nondecreasing in x. The bit string analogue of this is a fitness function $f : \{0, 1\}^n \to \mathbb{R}$ whose discrete derivative $\partial_i f(x) = f(x + e_i) - f(x)$ is nondecreasing in x for all $1 \leq i \leq n$ with e_i being the ith unit vector. A discrete function satisfying the aforementioned condition is called *submodular*. Submodularity is the counterpart of convexity in discrete settings (Lovász, 1983).

For understanding the properties of continuous optimizers it is central to study their performance for minimizing convex functions. This has been done in detail for continuous evolutionary algorithms (Beyer and Schwefel, 2002; Hansen, 2006). On the other hand, there is apparently very little prior work on the performance of discrete

A conference version of this article (Friedrich and Neumann, 2014) was presented at PPSN'14.

Manuscript received: November 3, 2014; revised: April 30, 2015; accepted: June 22, 2015.

evolutionary algorithms for optimizing submodular functions. The only reference we are aware of is by Rudolph (1996, sec. 5.1.2.3). He proves that there are submodular functions for which the $(1 + 1)$ EA requires exponential runtime (see also Bäck et al., 1997, sec. B2.4.2.5). We fill this gap and present several approximation results for simple evolutionary algorithms and submodular functions.

Analogous to the situation for convex functions, there is a significant difference between minimization and maximization of submodular functions. Submodular functions can be *minimized* with a (nontrivial) combinatorial algorithm in polynomial time (Iwata et al., 2001). On the other hand, submodular function *maximization* is NP-hard, as it generalizes many NP-hard combinatorial optimization problems, like maximum cut (Goemans and Williamson, 1995; Feige and Goemans, 1995), maximum directed cut (Halperin and Zwick, 2001), maximum facility location (Ageev and Sviridenko, 1999; Cornuejols et al., 1977), and several restricted satisfiability problems (Håstad, 2001; Feige and Goemans, 1995). As evolutionary algorithms are especially useful for hard problems, we focus on the maximization of submodular functions. Note that in general, submodular functions can also not be maximized *approximately* better than a constant factor unless P = NP (Feige, 1998).

More formally, we consider the optimization problem $\max\{f(S) : S \in \mathcal{I}\}$, where X is an arbitrary ground set, $f : 2^X \to \mathbb{R}$ is a fitness function, and $\mathcal{I} \subseteq 2^X$ is a collection of independent sets describing the feasible region of the problem. As usual, we assume *value oracle access* to the fitness function, i.e., for a given set S, an algorithm can query an oracle to find its value $f(S)$. We also always assume that the fitness function is normalized, i.e., $f(\emptyset) = 0$, and non-negative, i.e., $f(A) \geq 0$ for all $A \subseteq X$. We study the following variants of f and $\mathcal{I}$:

- *Submodular functions.* A function f is submodular iff $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq X$.
- *Monotone functions.* A function f is monotone iff $f(A) \leq f(B)$ for all $A \subseteq B$.
- *Symmetric functions.* A function f is symmetric iff $f(A) = f(X \setminus A)$ for all $A \subseteq X$.
- *Matroid.* A matroid is a pair $(X, \mathcal{I})$ composed of a ground set X and a nonempty collection $\mathcal{I}$ of subsets of X satisfying (1) If $A \in \mathcal{I}$ and $B \subseteq A$ then $B \in \mathcal{I}$ and (2) If $A, B \in \mathcal{I}$ and $|A| > |B|$ then $B + x \in \mathcal{I}$ for some $x \in A \setminus B$. The sets in $\mathcal{I}$ are called *independent*; the *rank* of a matroid is the size of any maximal independent set.
- *Uniform matroid.* A uniform matroid $(X, \mathcal{I})$ of rank $k \in \mathbb{N}$ contains all subsets of size at most k, i.e., $\mathcal{I} = \{A \subseteq X : |A| \leq k\}$.
- *Partition matroid.* A partition matroid is a matroid formed from a direct sum of uniform matroids. If the universe X is partitioned into k parts $X_1, \ldots, X_k$ and we have integers d_i with $0 \leq d_i \leq |X_i|$, then in a partition matroid a set I is independent if it contains at most d_i elements from each X_i, i.e., $|I \cap X_i| \leq d_i$ for all i. (Note that in the literature and in the conference version by Friedrich and Neumann (2014), it is assumed that $d_i = 1$ for all i.)
- *Intersection of k matroids.* Given k matroids $M_1 = (X, \mathcal{I}_1)$, $M_2 = (X, \mathcal{I}_2), \ldots, M_k = (X, \mathcal{I}_k)$ on the same ground set X, the intersection of these matroids is the matroid $(X, \mathcal{I})$ with $\mathcal{I} = \{A \subseteq X : A \in \mathcal{I}_i,\ 1 \leq i \leq k\}$. A simple example for

$k = 2$ is the family of matchings in a bipartite graph; or in general the family of hypergraph matchings in a k-partite hypergraph.

Maximizing submodular functions is not only NP-hard but also NP-hard to approximate. We therefore also have to formalize the notion of an approximation algorithm. We say an algorithm achieves an α-approximation if for all instances of the considered maximization problem, the output returned by the algorithm is at least α times the optimal value. In the context of evolutionary algorithms, we are interested in the expected time (usually measured by the number of fitness evaluations) until an evolutionary algorithm has achieved an α-approximation.

We study the well-known $(1 + 1)$ EA (Droste et al., 2002) as well as a multiobjective approach for optimizing submodular functions. Optimizing single objective optimization problems by multiobjective approaches such as the global simple evolutionary multiobjective optimizer (GSEMO) has already been shown to be beneficial for many single-objective optimization problems (Knowles et al., 2001; Jensen, 2004; Neumann and Wegener, 2006; Handl et al., 2008; Friedrich et al., 2010; Kratsch and Neumann, 2013). In this article, we prove the following statements.

- Based on the seminal work of Nemhauser et al. (1978), we show that the GSEMO achieves in polynomial time a $(1 - 1/e)$-approximation for maximizing *monotone submodular* functions under a *uniform matroid constraint* (Theorem 2). This approximation factor is optimal in the general setting (Nemhauser and Wolsey, 1978), and it is optimal even for the special case of Max-r-Cover, unless P = NP (Feige, 1998). Furthermore, we show that there are local optima for the $(1 + 1)$ EA that require exponential time to achieve an approximation better than $1/2 + \varepsilon$, for any $\varepsilon > 0$ a constant (Theorem 1).

- Based on work of Lee et al. (2010) and using the idea of p-exchanges, we show that the $(1 + 1)$ EA achieves a $(1/(k + 1/p + \varepsilon))$-approximation for any *monotone submodular* function f under k matroid constraints in expected time polynomial in $n^{\mathcal{O}(pk)}$ and $1/\varepsilon$, where $p \geq 1$ is an integer and $\varepsilon > 0$ is a real value (Theorem 3).

- Based on the work of Lee et al. (2009), we show that the GSEMO achieves in expected time $\mathcal{O}(\frac{1}{\varepsilon} \cdot n^{k+6} \log n)$ a $1/((k + 2)(1 + \varepsilon))$-approximation for maximizing *symmetric submodular* functions over *k matroid constraints*, where $\varepsilon > 0$ is a real value (Theorem 4). Furthermore, we explore the idea of p-exchanges and show that the GSEMO obtains (for $k \geq 2$, $p \geq 1$, and $\varepsilon > 0$) a $(\frac{1}{((1+\varepsilon)(k+1+1/p)})$-approximation in expected time $\mathcal{O}(\frac{1}{\varepsilon} \cdot n^{2p(k+1)+2} \cdot k \cdot \log n)$ (Theorem 5). Note that these results even hold for *nonmonotone* functions.

Friedrich and Neumann (2014) only studied the GSEMO. This article extends that work by providing lower and upper bounds for the $(1 + 1)$ EA (Section 3.1 and Section 4) as well as using the idea of p-exchanges to prove improved bounds for the GSEMO and the case of symmetric submodular functions in Section 5.

The paper is organized as follows. In Section 2 we describe the setting for submodular functions and introduce the algorithm that is the subject of our investigations. We analyze the algorithm on monotone submodular functions with a uniform constraint in Section 3 and present results for monotone submodular functions under k matroid constraints in Section 4. In Section 5 we consider the case of symmetric (but not necessarily monotone) submodular functions under k matroid constraints. Finally, we discuss open problems in Section 6.

2 Preliminaries

Optimization of submodular functions and matroids have received a lot of attention in the classical (nonevolutionary) optimization community. For a detailed exposition, we refer to the textbooks of Korte and Vygen (2007) and Schrijver (2003).

2.1 Submodular Functions and Matroids

When optimizing a submodular function $f : 2^X \to \mathbb{R}$, we often consider the incremental value of adding a single element. For this, we denote by $F_A(i) = f(A \cup \{i\}) - f(A)$ the marginal value of i with respect to A. Nemhauser et al. (1978, proposition 2.1) give seven equivalent definitions for submodular functions. Additionally to the definition stated in the introduction we also use that a function f is submodular iff $F_i(A) \geq F_i(B)$ for all $A \subseteq B \subseteq X$ and $i \in X \setminus B$.

Many common pseudo-Boolean and combinatorial fitness functions are submodular. As we are not aware of any general results for the optimization of submodular functions by evolutionary algorithms, we list a few examples of well-known submodular functions:

- *Linear functions.* All linear functions $f : 2^X \to \mathbb{R}$ with $f(A) = \sum_{i \in A} w_i$ for some weights $w : X \to \mathbb{R}$ are submodular. If $w_i \geq 0$ for all $i \in X$, then f is also monotone. Note that in the latter case, f is both submodular and supermodular and therefore modular.

- *Cut.* Given a graph $G = (V, E)$ with non-negative edge weights $w : E \to \mathbb{R}_{\geq 0}$. Let $S \subseteq V$ be a subset of vertices and $\delta(S)$ be the set of all edges that contain a vertex in S and $V \setminus S$. The cut function $w(\delta(S))$ is symmetric and submodular but not monotone.

- *Coverage.* Let the ground set be $X = \{1, 2, \ldots, n\}$. Given a universe U with n subsets $A_i \subseteq U$ for $i \in X$, and a non-negative weight function $w : U \to \mathbb{R}_{\geq 0}$. The coverage function $f : 2^X \to \mathbb{R}$ with $f(S) = |\bigcup_{i \in S} A_i|$ and the weighted coverage function f' with $f'(S) = w(\bigcup_{i \in S} A_i) = \sum_{u \in \bigcup_{i \in S} A_i} w(u)$ are monotone submodular.

- *Rank of a matroid.* The rank function $r(A) = \max\{|S| : S \subseteq A, S \in \mathcal{I}\}$ of a matroid $(X, \mathcal{I})$ is monotone submodular.

- *Hypervolume indicator.* Given a set of points in $\mathbb{R}^d$ in the objective space of a multiobjective optimization problem, measure the volume of the space dominated by these points relative to some fixed reference point. The hypervolume is a well-known quality measure in evolutionary multiobjective optimization and is known to be monotone submodular (Ulrich and Thiele, 2012).

We defined the most important matroids in the introduction. Matroid theory provides a framework in which many problems from combinatorial optimization can be studied from a unified perspective. Matroids are a special class of so-called *independence systems* that are given by a finite set X and a family of subsets $\mathcal{I} \subseteq X$ such that $\mathcal{I}$ is closed under subsets. Being a matroid is considered to be the property of an independence system that makes greedy algorithms work well. Within evolutionary computation, linear functions under matroid constraints have been considered by Reichel and Skutella (2010).

We assume a finite ground set $X = \{x_1, x_2, \ldots, x_n\}$ and identify each subset $S \subseteq X$ with a bit string $x \in \{0, 1\}^n$ such that the ith bit of x is 1 iff $x_i \in S$. Let $f : \{0, 1\}^n \rightarrow \mathbb{R}_{\geq 0}$ be the given submodular function and $F \subseteq \{0, 1\}^n$ be the set of feasible solutions. Note that f is defined on every element of $\{0, 1\}^n$. The constraints determining feasibility are given by k matroids, where k is a parameter. Given k arbitrary matroids $M_1, \ldots, M_k$ defined on a ground set X together with their independent systems $I_1, \ldots, I_k$. We consider the problem

$$\max \left\{ f(x) : x \in F := \bigcap_{j=1}^{k} I_j \right\},$$

where f is a submodular function defined on the ground set X.

Intersections of matroids occur in many settings like edge connectivity (Gabow, 1995), constrained minimum spanning trees (Hassin and Levin, 2004), and degree-bounded minimum spanning trees (Zenklusen, 2012).

A prominent example for matroid intersection constraints is the *maximum weight matching problem in bipartite graphs*: Given a bipartite graph $G = (V, E)$ with bipartition $V_1 \cup V_2$, let $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ be two partition matroids on E with

$$\mathcal{I}_1 = \{E' \subseteq E : |\delta(v) \cap E'| \leq 1, v \in V_1\},$$
$$\mathcal{I}_2 = \{E' \subseteq E : |\delta(v) \cap E'| \leq 1, v \in V_2\},$$

where $\delta(v)$ is the set of neighbors of v. Then it is easy to see that $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ if and only if I induces a matching in G.

Colorful spanning trees are an example for intersecting different kinds of matroids. Let $G = (V, E)$ with edges in E colored with k colors, that is, $E = E_1 \cup E_2 \cup \ldots \cup E_k$. Assume we are given integers $d_1, d_2, \ldots, d_k$ and aim at finding a spanning tree $T \subseteq E$ of G that has at most d_i edges of color i, i.e., $|T \cap E_i| \leq d_i$ for all i. Then this can be phrased as a matroid intersection problem, as it is the combination of a spanning tree matroid and a partition matroid.

2.2 Algorithms

The theoretical runtime analysis of evolutionary algorithms often considers randomized local search (RLS) and the (1 + 1) evolutionary algorithm (EA). We investigate the (1 + 1) EA (see Algorithm 1) and consider the fitness function $h(x) = (v(x), f(x))$, where $v(x)$ measures the constraint violation of x. Generalizing the fitness function used by Reichel and Skutella (2010) for the intersection of two matroids, we consider for problems with k matroid constraints $M_1, \ldots, M_k$,

$$v(x) = k \cdot |x|_1 - \sum_{j=1}^{k} r_j(x),$$

where $r_j(x)$ denotes the rank of x in matroid M_j, i.e.,

$$r_j(X) = \max\{|Y| : Y \subseteq X, Y \in I_j\}$$

for the set X given by x.

Algorithm 1 (1+1) EA algorithm

choose $x \in \{0,1\}^n$ uniformly at random
repeat
  create x' by flipping each bit x_i of x with probability $1/n$
  determine $h(x')$
  if $h(x') \geq h(x)$ **then**
    $x := x'$
until *stop*

Algorithm 2 GSEMO algorithm

choose $x \in \{0,1\}^n$ uniformly at random
determine $g(x)$
$P \leftarrow \{x\}$
repeat
  choose $x \in P$ uniformly at random
  create x' by flipping each bit x_i of x with probability $1/n$
  determine $g(x')$
  if x' is not strictly dominated by any other search point in P **then**
    include x' into P
    delete all other solutions $z \in P$ with $g(z) \leq g(x')$ from P
until *stop*

It is easy to see that $v(x) = 0$ iff x is a feasible solution and $v(x) > 0$ otherwise. We optimize $h(x)$ in lexicographic order, i.e.,

$$h(y) \geq h(x) \text{ holds iff } (v(y) < v(x)) \vee (v(y) = v(x) \wedge f(y) \geq f(x)).$$

We also consider a multiobjective approach to optimize submodular functions. The multiobjective counterparts of RLS and (1+1) EA are the simple evolutionary multi-objective optimizer (SEMO) (Laumanns et al., 2002) and global SEMO (GSEMO) (Giel, 2003). Both algorithms have been studied in detail (see Doerr et al., 2013; Giel, 2003; Giel and Lehre, 2010; Friedrich et al., 2010; Brockhoff et al., 2009). We consider the GSEMO given in Algorithm 2. For the multiobjective algorithm, we set $z(x) = f(x)$ iff $x \in F$ and $z(x) = -1$ iff $x \notin F$ and consider the multiobjective problem

$$g(x) := (z(x), |x|_0),$$

where $|x|_0 = \sum_{i=1}^{n}(1 - x_i)$ denotes the number of 0-bits in the given bit string x. We write $g(x) \geq g(y)$ iff $((z(x) \geq z(y)) \wedge (|x|_0 \geq |y|_0))$ holds. If $g(x) \geq g(y)$ holds, we say that y is dominated by x. The solution y is strictly dominated by solution x iff $g(x) \geq g(y)$ and $g(x) \neq g(y)$. A solution x that is not strictly dominated by any other solution $y \in X$

is called Pareto optimal, and the corresponding objective vector $g(x)$ is called Pareto optimal as well. The Pareto set of a given multiobjective problem consists of all Pareto optimal solutions, and the Pareto front consists of all Pareto optimal objective vectors. In our studies, we focus on the solution $x^* = \arg\max_{x \in P} z(x)$ of the GSEMO and study the quality of this solution.

We study the expected number of iterations (of the repeat loop) of the $(1+1)$ EA and the GSEMO until their feasible solution x^* is for the first time an α-approximation of an optimal feasible solution OPT, i.e., $f(x^*)/\text{OPT} \geq \alpha$ holds. Here α denotes the investigated approximation ratio for the considered problem. We call the expected number of iterations to reach an α-approximation, the expected (run)time to achieve an α-approximation.

3 Monotone Submodular Functions with a Uniform Constraint

In this section, we investigate submodular functions with one uniform constraint. In the case of one uniform constraint of size r, a solution $x \in X$ is feasible if it has at most r elements. Hence, we have $F = \{x : x \in X \wedge |x|_1 \leq r\}$.

3.1 Lower bound for the $(1+1)$ EA

We consider the $(1+1)$ EA and show that this approach has to cope with local optima with a large inferior neighborhood. Getting trapped in these local optima, the algorithm finds it hard to achieve an approximation ratio greater than $1/2 + \varepsilon$, where $\varepsilon > 0$ is a constant.

Based on our previously defined fitness function, we have $v(x) = \max\{0, |x|_1 - r\}$, as we are considering problems with one uniform constraint. To show the upper bound on the approximation ratio, we consider an instance of the Max-r-Cover problem.

Our instance is obtained from a bipartite graph that has already been investigated in the context of the vertex cover problem (Friedrich et al., 2010). Let $G = (V_1 \cup V_2, E)$ be the complete bipartite graph on $V_1 = \{v_1, \ldots, v_{\varepsilon n}\}$ and $V_2 = \{v_{\varepsilon n+1}, \ldots, v_n\}$, where $|V_1| = \varepsilon n$ and $|V_2| = (1-\varepsilon)n$ for $\varepsilon < 0.1$. The ground set is given by the set of edges E, and each node $v_i \in V_1 \cup V_2$ is identified with the subset of edges adjacent to v_i, i.e., $E_i = \{e \in E : e \cap v_i \neq \emptyset\}$. Let E^{V_1} and E^{V_2} be the set of subsets corresponding to the nodes of V_1 and V_2, respectively. We consider the $(1+1)$ EA working with bit strings of length n, where the set E_i is chosen iff $x_i = 1$, $1 \leq i \leq n$.

For the constraint, we set $r = (1 - 2\varepsilon + \delta) \cdot n$, where δ, $0 < \delta < \varepsilon$, is an arbitrary small positive constant as an upper bound on the number of sets. Furthermore, we require $\varepsilon n \leq r$, which is equivalent to $1 - 2\varepsilon + \delta \leq \varepsilon$. This implies that E^{V_1} is an optimal solution covering the whole ground set E and can be achieved by setting $1/2 > \varepsilon \geq (1-\delta)/3$.

We consider the solution x^ℓ where r subsets of E^{V_2} and no subset of E^{V_1} is selected. The value of an optimal solution is $\text{OPT} = \varepsilon(1-\varepsilon)n^2/2$ and we have $f(x^\ell) = r \cdot (\varepsilon n)/2$. The approximation ratio of x^ℓ is $\alpha(x^\ell) = f(x^\ell)/\text{OPT} = (1 - 2\varepsilon + \delta)/(1-\varepsilon)$. Setting $\varepsilon = (1-\delta)/3$ we get

$$\begin{aligned}
\alpha(x^\ell) &= (1 - 2(1/3 - \delta/3) + \delta)/(1 - 1/3 + \delta/3) \\
&= (1/3 + 5\delta/3)/(2/3 + \delta/3) \\
&= (1 + 5\delta)/(2 + \delta).
\end{aligned}$$

Choosing δ as a constant arbitrary close to 0, this expression becomes a constant arbitrary close to 1/2.

THEOREM 1: *There are monotone submodular functions f for which the (1 + 1) EA under a uniform matroid constraint may end up in bad local optima. More precisely, there is an instance of the Max-r-Cover problem such that starting with* x^{ℓ}*, the expected waiting time for the (1 + 1) EA to achieve an improvement and therefore a solution with an approximation ratio greater than* $(1 + 5\delta)/(2 + \delta)$ *is* $e^{\Omega(n)}$.

PROOF: The search point x^{ℓ} has r chosen elements, and inserting any further elements without removing any other elements is not accepted. Furthermore, removing one or more elements without inserting any new ones covers fewer elements, which is therefore also not accepted. Each selected set of E^{V_2} covers εn elements that are not covered by any other chosen element, whereas each set of E^{V_1} would gain an additional contribution of at most $(1 - \varepsilon - (1 - 2\varepsilon + \delta)) = \varepsilon - \delta$ elements.

In order to have a set of E^{V_1} included and accepted, at least δn chosen sets of E^{V_2} have to be removed. Removing δn such elements decreases the fitness by $\delta \varepsilon n^2/2$ and has to be compensated by choosing at least δn sets of E^{V_1}.

Hence, in order to have a new accepted solution, $2\delta n$ bits have to flip in a mutation step. The probability for this is at most

$$\binom{\varepsilon n}{\delta n} \cdot \binom{(1 - \varepsilon)n}{\delta n} \cdot \left(\frac{1}{n}\right)^{2\delta n} = e^{-\Omega(n)}.$$

This implies an expected waiting time of $e^{\Omega(n)}$ and therefore completes the proof. □

We conjecture that the previous theorem may be generalized to initial solutions chosen uniformly at random by following the analysis of the (1+1) EA for the vertex cover problem on complete bipartite graphs (Friedrich et al., 2010). We do not carry out such a technical analysis, as our purpose in this section is to point out a situation where the (1+1) EA gets stuck in a local optimum with an approximation ratio roughly 1/2 of the Max-*r*-Cover problem.

3.2 Upper Bound for the GSEMO

We now turn to the GSEMO and show that this approach does not have to cope with local optima that may prevent the algorithm from achieving an approximation ratio better than 1/2. The GSEMO has the ability of carrying out local search operations but also allows for a greedy behavior, which is beneficial in this case. The greedy behavior of the GSEMO leads to the following result.

THEOREM 2: *The expected time until the GSEMO has obtained a* $(1 - \frac{1}{e})$*-approximation for a monotone submodular function f under a uniform constraint of size r is* $\mathcal{O}(n^2 (\log n + r))$.

PROOF: We first study the expected time until the GSEMO has produced the solution 0^n for the first time. This solution is Pareto optimal and will therefore stay in the population after it has been produced for the first time. Furthermore, the population size is upper bounded by $n + 1$, as it contains for each i, $0 \le i \le n$ at most one solution having exactly i 1-bits. The solution 0^n is feasible and has the maximum number of 0-bits. This implies that the population will not include any infeasible solution to the submodular function f after having included 0^n.

For this step, we consider in each iteration the individual y that has the minimum number of 1-bits among all individuals in the population and denote $\ell = |y|_1$ the number of 1-bits in this individual. Note, that ℓ cannot increase during the run of the algorithm. For $1 < \ell \le n$ a solution y' with $|y'|_1 = \ell - 1$ is produced with probability at least $\ell/(en^2)$, as y' can be produced by selecting y for mutation and flipping one of the ℓ 1-bits. The expected waiting time to include the solution 0^n for the first time into the population is therefore upper bounded by $\sum_{\ell=1}^{n}\left(\frac{\ell}{en^2}\right)^{-1} = \mathcal{O}(n^2 \log n)$.

For the remainder of the proof, we follow the ideas of the proof for the greedy algorithm in Nemhauser et al. (1978). We show that the GSEMO produces in expected time $\mathcal{O}(n^2 k)$ for each $0 \leq j \leq r$ a solution X_j with

$$f(X_j) \geq \left(1-\left(1-\frac{1}{r}\right)^j\right) \cdot f(\text{OPT}), \tag{1}$$

where $f(\text{OPT})$ denotes the value of a feasible optimal solution. Note, that a solution is feasible iff it has at most r 1-bits. After having included the solution 0^n into the population, this is true for $j = 0$. The proof is done by induction. Assume that the GSEMO has already obtained a solution fulfilling Equation (1) for each j, $0 \leq j \leq i < r$. We claim that choosing the solution $x \in P$ with $|x|_1 = i$ for mutation and inserting the element corresponding to the largest possible increase of f increases the value of f by at least $\delta_{i+1} \geq \frac{1}{r} \cdot (f(\text{OPT}) - f(X_i))$. Let δ_{i+1} be the increase in f that we obtain when choosing the solution $x \in P$ with $|x|_1 = i$ for mutation and inserting the element corresponding to the largest possible increase.

Because of monotonicity and submodularity, we have $f(\text{OPT}) \leq f(X_i \cup \text{OPT}) \leq f(X_i) + r\delta_{i+1}$, which implies $\delta_{i+1} \geq \frac{1}{r} \cdot (f(\text{OPT}) - f(X_i))$. This leads to

$$f(X_{i+1}) \geq f(X_i) + \frac{1}{r}(f(\text{OPT}) - f(X_i)) \geq \left(1-\left(1-\frac{1}{r}\right)^{i+1}\right) \cdot f(\text{OPT}).$$

For $i = r$, we get $(1-(1-\frac{1}{r})^r) \cdot f(\text{OPT}) \geq (1-\frac{1}{e}) f(\text{OPT})$. The probability for such a step going from i to $i+1$ is lower bounded by $\frac{1}{en^2}$ and hence the expected time until a $(1-\frac{1}{e})$-approximation has been obtained is at most

$$\mathcal{O}(n^2 \log n) + \sum_{i=0}^{r}\left(\frac{1}{en^2}\right)^{-1} = \mathcal{O}(n^2(\log n + r)). \qquad \square$$

We demonstrate the applicability of Theorem 2 by two examples. First, consider the maximum coverage problem introduced in Section 2. Given a universe U with subsets $A_1, A_2, \ldots, A_n \subseteq U$, we want to maximize a coverage function $f(S) = |\bigcup_{i \in S} A_i|$ such that $|S| \leq r$. Theorem 2 immediately implies the following.

COROLLARY 1: *The expected time until the GSEMO has obtained a $(1-1/e)$-approximation for the Max-r-Cover problem is $\mathcal{O}(n^2(\log n + r))$. The achieved approximation factor is optimal, unless $P = NP$ (Feige, 1998).*

As a second example, we consider a problem from evolutionary multiobjective optimization. As discussed in Section 2, the hypervolume indicator is a monotone submodular function. The hypervolume subset selection problem (HYP-SSP), where we are given n points in $\mathbb{R}^d$ and want to select a subset of size k with maximal hypervolume, therefore aims at maximizing a monotone submodular function $f : \{0,1\}^n \to \mathbb{R}_{\geq 0}$ under a uniform matroid constraint of rank k. HYP-SSP has been addressed by a number of authors (Bringmann et al., 2014a; 2014b; Glasmachers, 2014; Kuhn et al., 2014; Guerreiro et al., 2015). Theorem 2 has the following implication for HYP-SSP.

COROLLARY 2: *The expected time until the GSEMO has obtained a $(1-1/e)$-approximation for HYP-SSP is $\mathcal{O}(n^2(k + \log n))$.*

For dimensions $d > 2$ this is significantly faster than the best known exact algorithm with runtime $\mathcal{O}(n^k)$ (Bringmann and Friedrich, 2010). Note that HYP-SSP can be solved in time $\mathcal{O}(n\,(k + \log n))$ for $d = 2$ (Bringmann et al., 2014b).

4 Monotone Submodular Functions under Matroid Constraints

The previous section only studied uniform matroid constraints. We now extend this to general matroids and intersection of k matroids, and study monotone submodular functions under constraints given by k matroids $M_1, \ldots, M_k$.

We consider the (1 + 1) EA and start by analyzing the time until the algorithm has obtained a feasible solution x with $f(x) \geq \text{OPT}/n$. This result serves as the basis for the main result of this section.

LEMMA 1: *Let f be a monotone submodular function under $k \geq 1$ matroid constraints and* OPT *be the value of an optimal solution. The expected time until the (1 + 1) EA has obtained a feasible solution with $f(x) \geq$* OPT*$/n$ is $\mathcal{O}(n^{k+1})$.*

PROOF: The (1 + 1) EA starts with the initial solution chosen uniformly at random. We first consider the expected time until the algorithm has obtained for the first time a feasible solution, i.e., a solution x for which $v(x) = 0$ holds. To do this, we generalize Proposition 10 of Reichel and Skutella (2010) to the case of the intersection of k matroids. Suppose that x is an infeasible solution with $\ell = v(x)$. During the optimization process, ℓ never increases, and there are at least ℓ/k distinct elements that can be removed to decrease ℓ. Hence, the probability of decreasing ℓ is at least

$$\frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{\ell}{ekn},$$

and the expected time until a feasible solution has been produced is upper bounded by

$$\sum_{\ell=1}^{kn} \frac{ekn}{\ell} = \mathcal{O}(kn\,(\log k + \log n)).$$

For the remainder of the proof, we work under the assumption that a feasible solution has already been obtained. Let x be an arbitrary feasible solution and x^* be an optimal solution. Furthermore, let a be the element in x^* such that $f(\{a\}) \geq \text{OPT}/n$. As f is monotone, we have $f(y) \geq \text{OPT}/n$ for any feasible solution containing the element a. According to Theorem 2.1 of Lee et al. (2009), a feasible solution y containing a can be obtained from any feasible solution x by introducing a and removing at most k elements from x. The expected waiting time of the (1 + 1) EA for such a $(k + 1)$-bit flip is $\mathcal{O}(n^{k+1})$. Altogether, the expected time to produce a feasible solution x with $f(x) \geq \text{OPT}/n$ is $\mathcal{O}(n^{k+1})$, as $\mathcal{O}(kn\,(\log k + \log n)) = \mathcal{O}(n^{k+1})$ for any $k \geq 1$. □

In the previous section, we showed that there are local optima for submodular functions with one uniform constraint that only constitute an approximation ratio of at most $1/2 + \delta$. Furthermore, the (1 + 1) EA requires exponential time to leave these local optima. The following theorem shows that the (1 + 1) EA obtains a $1/(k + \delta)$-approximation for any constant $k \geq 2$ and δ in expected polynomial time. For the case $k = 1$, this implies a $1/(2 + \delta)$-approximation in expected polynomial time, as we may duplicate the single matroid constraining the search space.

THEOREM 3: *For any integers $k \geq 2$, $p \geq 1$ and real value $\varepsilon > 0$, the expected time until the (1 + 1) EA has obtained a $(1/(k + 1/p + \varepsilon))$-approximation for any monotone submodular function f under k matroid constraints is $\mathcal{O}(\frac{1}{\varepsilon} \cdot n^{2p(k+1)+1} \cdot k \cdot \log n)$.*

PROOF: Because of Lemma 1, a feasible solution x with $f(x) \geq \text{OPT}/n$ is obtained in expected time $\mathcal{O}(n^{k+1})$. In the following, we work under the assumption that the algorithm has obtained a feasible solution x with $f(x) \geq \text{OPT}/n$. A p-exchange operation applied to the current solution x introduces at most $2p$ new elements and deletes at most $2kp$ elements of x. A solution y that can be obtained from x by a p-exchange operation is called a p-exchange neighbor of x. According to Lee et al. (2010), every solution x for which there exists no p-exchange neighbor y with $f(y) \geq (1 + \frac{\varepsilon}{n(k+1)}) \cdot f(x)$ is a $(1/(k + 1/p + \varepsilon))$-approximation for any monotone submodular function.

The expected waiting time for a specific p-exchange operation is $\mathcal{O}(n^{2p(k+1)})$, as the probability for a specific p-exchange is $\Omega(n^{-2p(k+1)})$. The number of steps producing from a solution x, a solution y with $f(y) \geq \left(1 + \frac{\varepsilon}{n(k+1)}\right) \cdot f(x)$ is at most

$$\log_{1+\frac{\varepsilon}{n(k+1)}} \frac{\text{OPT}}{\text{OPT}/n} = \mathcal{O}\left(\frac{1}{\varepsilon}\, n\,(k+1) \log n\right).$$

Altogether, the expected time until the (1 + 1) EA has obtained a $(1/(k + 1/p + \varepsilon))$-approximation is $\mathcal{O}(\frac{1}{\varepsilon} \cdot n^{2p(k+1)+1} \cdot k \cdot \log n)$. □

Recall the example of finding colorful spanning trees (Section 2), which can be described as a monotone submodular maximization problem under $k = 2$ matroid constraints. By choosing $p > 1/\varepsilon$ sufficiently large, we get the following corollary.

COROLLARY 3: *The expected time until the (1 + 1) EA has obtained a $(1/2 - \varepsilon)$-approximation for colorful spanning trees is $\mathcal{O}(\text{poly}(n)/\varepsilon)$ for all $\varepsilon > 0$.*

5 Symmetric Submodular Functions under Matroid Constraints

We now turn to symmetric submodular functions that are not necessarily monotone. For our analysis, we make use of the following corollary, which can be obtained from Lee et al. (2009).

COROLLARY 4: *Let x be a solution such that no solution with fitness at least $\left(1 + \frac{\varepsilon}{n^4}\right) \cdot f(x)$ can be achieved by deleting one element, or by inserting one element and deleting at most k elements. Then x is a $(\frac{1}{(k+2)(1+\varepsilon)})$-approximation.*

Corollary 4 states that there is always the possibility of achieving a certain progress if no good approximation has been obtained. We use this to show the following results for the GSEMO. It should be noted that the corresponding Theorem 2 in Friedrich and Neumann (2014) is accidentally missing the symmetry condition.

THEOREM 4: *The expected time until the GSEMO has obtained a $(\frac{1}{(k+2)(1+\varepsilon)})$-approximation for any symmetric submodular function under k matroid constraints is $\mathcal{O}(\frac{1}{\varepsilon} n^{k+6} \log n)$.*

PROOF: Following previous investigations, the GSEMO introduces the solution 0^n in the population after an expected number of $\mathcal{O}(n^2 \log n)$ steps. This solution is Pareto optimal and will from that point on stay in the population. Furthermore, 0^n is a feasible solution and has the largest possible number of 0-bits. Hence, from the time 0^n has been included in the population, the population will never include infeasible solutions.

Selecting 0^n for mutation and inserting the element that leads to the largest increase in the f-value produces a solution y with $f(y) \geq \text{OPT}/n$. The reason for this is that the number of elements is limited by n and that f is submodular. Having obtained a solution of fitness at least OPT/n, we focus in each iteration on the individual having the largest f-value in P. Because of the selection mechanism of the GSEMO, a solution with the maximal f-value will always stay in the population, and the value will not decrease during the run of the algorithm.

As long as the algorithm has not obtained a solution of the desired quality, it can produce from its solution x with the highest f-value a feasible offspring y such that $f(y) \geq \left(1 + \frac{\varepsilon}{n^4}\right) \cdot f(x)$. The expected waiting time for this event is $\mathcal{O}(n^{k+2})$, as at most $k+1$ specific bits of x have to be flipped and using the fact that the population size is at most $n+1$.

Starting with a solution of quality at least OPT/n, the number of such steps in order to achieve an optimal solution is upper bounded by

$$\log_{(1+\frac{\varepsilon}{n^4})} \frac{\mathrm{OPT}}{\mathrm{OPT}/n} = \mathcal{O}\left(\frac{1}{\varepsilon} n^4 \log n\right).$$

Hence, the expected time to achieve a $(\frac{1}{(k+2)(1+\varepsilon)})$-approximation is $\mathcal{O}(\frac{1}{\varepsilon} n^{k+6} \log n)$. □

As an example, let us consider again the NP-hard maximum cut problem, where for a given graph $G = (V, E)$ with n vertices and non-negative edge weights $w : E \to \mathbb{R}_{\geq 0}$. We want to maximize the cut function $\delta(S)$ over all $S \subseteq V$ as defined in Section 2. It is known that the greedy algorithm achieves a 0.5-approximation, while the best known algorithms achieve a 0.87856-approximation (Goemans and Williamson, 1995). Theorem 4 immediately implies the following.

COROLLARY 5: *The expected time until the GSEMO has obtained a $1/(3\,(1+\varepsilon))$-approximation for the maximum cut problem is $\mathcal{O}(\frac{1}{\varepsilon} n^7 \log n)$.*

Note that this result is presumably not tight. We conjecture that a less general analysis can show that the GSEMO achieves a 1/2-approximation.

Using the idea of p-exchanges from Theorem 3, we can improve the approximation result of Theorem 4 with an increasing runtime depending on p.

THEOREM 5: *For any integers $k \geq 2$, $p \geq 1$ and real value $\varepsilon > 0$, the expected time until the GSEMO has obtained a $(\frac{1}{(1+\varepsilon)(k+1+1/p)})$-approximation for any symmetric submodular function under k matroid constraints is $\mathcal{O}(\frac{1}{\varepsilon} \cdot n^{2p(k+1)+2} \cdot k \cdot \log n)$.*

PROOF: The GSEMO produces a feasible solution x with $f(x) \geq \mathrm{OPT}/n$ in expected time $\mathcal{O}(n^2 \log n)$ (see proof of Theorem 4). After the GSEMO has obtained a solution x with $f(x) \geq \mathrm{OPT}/n$, we focus on the solution with the largest f-value in the population.

According to Lemma 3.2 of Lee et al. (2010) for $k \geq 2$, we have

$$(1+\varepsilon)(k+1/p) \cdot f(S) \geq f(C \cup S) + (k-1+1/p) f(S \cap C) \geq f(S \cup C) + f(S \cap C)$$

if there is no p-exchange neighbor T with $f(T) \geq (1 + \frac{\varepsilon}{n(k+1)}) \cdot f(S)$. As f is symmetric, we have $f(S) = f(X \setminus S)$, and adding $f(X \setminus S)$ to both sides yields

$$(1+\varepsilon)(k+1/p+1) f(S) \geq f(X \setminus S) + f(S \cup C) + f(S \cap C) \geq f(C),$$

which implies $f(S)/f(C) \geq 1/((1+\varepsilon)(k+1+1/p))$. The number of improvements by a factor $(1 + \frac{\varepsilon}{n(k+1)})$ is upper bounded by

$$\log_{1+\frac{\varepsilon}{n(k+1)}} n = \mathcal{O}\left(\frac{1}{\varepsilon} n(k+1) \log n\right).$$

Furthermore, the expected waiting time for such an improvement is $\mathcal{O}(n^{2p(k+1)+1})$, as the population size is upper bound by $n+1$ and a specific p-exchange has probability $\Omega(n^{-2p(k+1)})$ (see proof of Theorem 3). This completes the proof. □

6 Discussion and Open Problems

Maximizing submodular functions under matroid constraints is a very general optimization problem that contains many classical combinatorial optimization problems like maximum cut (Goemans and Williamson, 1995; Feige and Goemans, 1995), maximum directed cut (Halperin and Zwick, 2001), maximum facility location (Ageev and Sviridenko, 1999; Cornuejols et al., 1977), and others. We presented several positive and negative results for the approximation behavior of the simple evolutionary algorithms in the framework. To the best of our knowledge, this is the first paper on the analysis of evolutionary algorithms optimizing *submodular functions*. The only result on the performance of evolutionary algorithms under *matroid constraints* is by Reichel and Skutella (2010). They showed that the $(1+1)$ EA achieves in polynomial time a $1/k$-approximation for maximizing a linear function subject to k matroid constraints.

This paper gives a first set of results but also raises many new questions. We briefly name a few:

- We only study the $(1+1)$ EA and SEMO algorithms, but similar results might be possible for population-based algorithms with appropriate diversity measures.
- Our runtime upper bounds might not be tight. It would be interesting to show matching lower bounds, especially for comparing different algorithms and function classes.
- The proven approximation guarantees hold for very general problem classes. Much tighter results should be possible for specific problems like maximum cut.
- Minimizing submodular functions is in general simpler than maximizing submodular functions. However, it is not obvious what this implies for evolutionary algorithms minimizing submodular functions.
- Our proofs strongly rely on the greedy-like behavior of SEMO. It might be possible to prove a general relationship between SEMO and greedy algorithms; or to give an example where SEMO strictly outperforms a greedy strategy.
- We assume value oracle access to the fitness function f. It might be worth studying the black box complexity of submodular functions in the sense of Lehre and Witt (2012).
- We studied submodular fitness functions that are either monotone or symmetric. Future work should also cover submodular functions that are neither monotone nor symmetric.

Acknowledgments

The research leading to these results received funding from the Australian Research Council (ARC) under grant agreement DP140103400 and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 618091 (SAGE).

References

Ageev, A. A., and Sviridenko, M. (1999). An 0.828-approximation algorithm for the uncapacitated facility location problem. *Discrete Applied Mathematics*, 93(2-3): 149–156.

Bäck, T., Fogel, D. B., and Michalewicz, Z. (Eds.) (1997). *Handbook of evolutionary computation*. Oxford: Oxford University Press.

Beyer, H.-G., and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, 1(1): 3–52.

Bringmann, K., and Friedrich, T. (2010). An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3): 383–402.

Bringmann, K., Friedrich, T., and Klitzke, P. (2014a). Generic postprocessing via subset selection for hypervolume and epsilon-indicator. In *Parallel Problem Solving from Nature*, pp. 518–527. Lecture Notes in Computer Science, Vol. 8672.

Bringmann, K., Friedrich, T., and Klitzke, P. (2014b). Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 589–596.

Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., and Zitzler, E. (2009). On the effects of adding objectives to plateau functions. *IEEE Transactions on Evolutionary Computation*, 13(3): 591–603.

Cornuejols, G., Fisher, M., and Nemhauser, G. L. (1977). On the uncapacitated location problem. In P. L. Hammer, E. L. Johnson, B. H. Korte, and G. M. Nemhauser (Eds.), *Studies in integer programming*, pp. 163–177. Amsterdam: Elsevier.

Doerr, B., Kodric, B., and Voigt, M. (2013). Lower bounds for the runtime of a global multi-objective evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 432–439.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1-2): 51–81.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4): 634–652.

Feige, U., and Goemans, M. X. (1995). Approximating the value of two power proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings of the Israel Symposium on Theory and Computing Systems*, pp. 182–189.

Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., and Witt, C. (2010). Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4): 617–633.

Friedrich, T., and Neumann, F. (2014). Maximizing submodular functions under matroid constraints by multi-objective evolutionary algorithms. In *Parallel Problem Solving from Nature*, pp. 922–931. Lecture Notes in Computer Science, Vol. 8672.

Gabow, H. N. (1995). A matroid approach to finding edge connectivity and packing arborescences. *Journal of Computer and System Science*, 50(2): 259–273.

Giel, O. (2003). Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1918–1925.

Giel, O., and Lehre, P. K. (2010). On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18(3): 335–356.

Glasmachers, T. (2014). Optimized approximation sets of low-dimensional benchmark Pareto fronts. In *Parallel Problem Solving from Nature*, pp. 569–578. Lecture Notes in Computer Science, Vol. 8672.

Goemans, M. X., and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6): 1115–1145.

Guerreiro, A. P., Fonseca, C. M., and Paquete, L. (2015). Greedy hypervolume subset selection in the three-objective case. In *Proceedings of the Conference on Genetic and Evolutionary Computation Conference (GECCO)*, pp. 671–678.

Halperin, E., and Zwick, U. (2001). Combinatorial approximation algorithms for the maximum directed cut problem. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 1–7.

Handl, J., Lovell, S. C., and Knowles, J. D. (2008). Multiobjectivization by decomposition of scalar cost functions. In *Parallel Problem Solving from Nature*, pp. 31–40. Lecture Notes in Computer Science, Vol. 5199.

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea (Eds.), *Towards a new evolutionary computation: Advances in estimation of distribution algorithms*, pp. 75–102. Berlin: Springer.

Hassin, R., and Levin, A. (2004). An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2): 261–268.

Håstad, J. (2001). Some optimal inapproximability results. *Journal of the ACM*, 48(4): 798–859.

Iwata, S., Fleischer, L., and Fujishige, S. (2001). A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4): 761–777.

Jensen, M. T. (2004). Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal Mathematical Modelling and Algorithms*, 3(4): 323–347.

Knowles, J. D., Watson, R. A., and Corne, D. (2001). Reducing local optima in single-objective problems by multi-objectivization. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, pp. 269–283.

Korte, B., and Vygen, J. (2007). *Combinatorial optimization: Theory and algorithms*, 4th ed. Berlin: Springer.

Kratsch, S., and Neumann, F. (2013). Fixed-parameter evolutionary algorithms and the vertex cover problem. *Algorithmica*, 65(4): 754–771.

Kuhn, T., Fonseca, C. M., Paquete, L., Ruzika, S., and Figueira, J. R. (2014). *Hypervolume subset selection in two dimensions: Formulations and algorithms.* (Technical Report.) Fachbereich Mathematik, TU Kaiserslautern.

Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., and Deb, K. (2002). Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature*, pp. 44–53. Lecture Notes in Computer Science, Vol. 2439.

Lee, J., Mirrokni, V. S., Nagarajan, V., and Sviridenko, M. (2009). Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 323–332.

Lee, J., Sviridenko, M., and Vondrák, J. (2010). Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operational Research*, 35(4): 795–806.

Lehre, P. K., and Witt, C. (2012). Black-box search by unbiased variation. *Algorithmica*, 64(4): 623–642.

Lovász, L. (1983). Submodular functions and convexity. In A. Bachem, B. Korte, and M. Grötschel, (Eds.), *Mathematical programming: The state of the art*. Berlin: Springer.

Nemhauser, G., Wolsey, L., and Fisher, M. (1978). An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1): 265–294.

Nemhauser, G. L., and Wolsey, L. A. (1978). Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3): 177–188.

Neumann, F., and Wegener, I. (2006). Minimum spanning trees made easier via multiobjective optimization. *Natural Computing*, 5(3): 305–319.

Reichel, J., and Skutella, M. (2010). Evolutionary algorithms and matroid optimization problems. *Algorithmica*, 57(1): 187–206.

Rudolph, G. (1996). *Convergence properties of evolutionary algorithms*. (Unpublished doctoral dissertation.) Universität Dortmund.

Schrijver, A. (2003). *Combinatorial optimization: Polyhedra and efficiency*. Berlin: Springer.

Ulrich, T., and Thiele, L. (2012). Bounding the effectiveness of hypervolume-based $(\mu + \lambda)$-archiving algorithms. In *Proceedings of the International Conference on Learning and Intelligent Optimization*, pp. 235–249.

Zenklusen, R. (2012). Matroidal degree-bounded minimum spanning trees. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1512–1521.