On the Size of Weights in Randomized Search Heuristics *

Joachim Reichel
Institut für Mathematik
TU Berlin
Berlin, Germany
reichel@math.tu-berlin.de

Martin Skutella
Institut für Mathematik
TU Berlin
Berlin, Germany
skutella@math.tu-berlin.de

ABSTRACT

Runtime analyses of randomized search heuristics for combinatorial optimization problems often depend on the size of the largest weight. We consider replacing the given set of weights with smaller weights such that the behavior of the randomized search heuristic does not change. Upper bounds on the size of the new, equivalent weights allow us to obtain upper bounds on the expected runtime of such randomized search heuristics independent of the size of the actual weights. Furthermore we give lower bounds on the largest weights for worst-case instances. Finally we present some experimental results, including examples for worst-case instances.

Categories and Subject Descriptors: F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity

General Terms: Theory, Algorithms, Performance

Keywords: randomized search heuristics, evolutionary algorithms

1. INTRODUCTION

We consider combinatorial optimization problems on the search space $S = \{0,1\}^n$. The set of feasible search points is denoted by $F \subseteq S$. For simplification, we restrict ourselves to minimization problems. The objective function $f : S \mapsto \mathbb{Z}$ is given by $f(x) = \sum_{i=1}^{n} W_i x_i$ for $x \in F$ with integral positive weights $W_i \in \mathbb{N}$. We demand that f separates F and $S \setminus F$, i. e., $f(x) < f(y)$ for all $x \in F$ and $y \in S \setminus F$. We also assume that the feasibility of a search point $x \in S$ does not depend on the weights W_i. In other words, the set F of feasible search points is independent from the weights W_i. Let $H(x, y)$ denote the Hamming distance of $x, y \in S$.

We consider the following class of randomized search heuristics [8, 9].

Algorithm 1. *Randomized Search Heuristic (RSH$_\ell$)*
1. *Choose $x \in F$.*
2. *Repeat*
   - *Choose $x' \in S$ such that $H(x, x') \le \ell$.*
   - *If $f(x') \le f(x)$, then $x \leftarrow x'$.*

In each step, RSH$_\ell$ chooses a search point x' from the neighborhood of the current search point x that consists of all search points in S with a Hamming distance of at most ℓ. The acceptance of x' is only based on the sign of $f(x') - f(x)$, not on the value $f(x') - f(x)$ itself. The variant RSH$^*_\ell$ of RSH$_\ell$ accepts search points x' if and only if $f(x') < f(x)$.

We do not make any assumptions on the way x and x' are chosen. A well-studied evolutionary algorithm called (1+1) EA obtains x' by flipping the bits of x with probability $1/n$. Another evolutionary algorithms called RLS$_\ell$ flips up to ℓ bits according to a fixed probability distribution, where ℓ is typically a small number, e. g., $\ell = 2$ or $\ell = 3$. Some local search algorithms consider the entire neighborhood within Hamming distance ℓ and pick x' from this neighborhood according to some criterion. Tabu search methods maintain a set of forbidden search points that are *not* considered in the current iteration.

Note that in our description of Algorithm 1 the initial search point x is chosen from the set F of feasible search points. Often one chooses the initial search point randomly from the search space S such that the algorithm does not necessarily start from a feasible solution. In this case, divide the run of RSH$_\ell$ into two phases. The second phase starts as soon as a feasible search point $x \in F$ has been found. By definition of the objective function f, infeasible search points are never accepted in the second phase. Then our results apply to the analysis of the second phase.

Since runtime analyses of such randomized search heuristics often depend on the largest weight $W_{\max}$ [2, 4, 5, 6, 7], we would like to replace the weights $W_1, \dots, W_n$ by new weights $w_1, \dots, w_n$ such that $w_{\max}$ is as small as possible under the condition that the behavior of RSH$_\ell$ does not change.

In particular, we would like to bound the minimal $w_{\max}$ from above over all inputs W_i. In the runtime analysis, such an upper bound can be used instead of $W_{\max}$. Note that the

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
FOGA'09, January 9–11, 2009, Orlando, Florida, USA.
Copyright 2009 ACM 978-1-60558-414-0/09/01 ...$5.00.

problem	known result depending on $W_{\max}$	algorithm	ℓ	upper bd. on $w_{\max}$	new result independent of $W_{\max}$
Minimum Spanning Tree [6]	$O(\|E\|^2(\log\|V\| + \log W_{\max}))$	RLS (1+1) EA	2 $\|E\|$	$\|E\|$ $\|E\|^{\|E\|/2}$	$O(\|E\|^2 \log\|V\|)$ $O(\|E\|^3 \log\|V\|)$
Minimum Weight Basis [7]	$O(\|E\|^2(\log r(E) + \log W_{\max}))$	RLS (1+1) EA	2 $\|E\|$	$\|E\|$ $\|E\|^{\|E\|/2}$	$O(\|E\|^2 \log\|E\|)$ $O(\|E\|^3 \log\|E\|)$
Weighted Matroid Intersection [7]	$O(\|E\|^4(\log r(E) + \log W_{\max}))$	RLS_3 (1+1) EA	3 $\|E\|$	$2^{\|E\|}$ $\|E\|^{\|E\|/2}$	$O(\|E\|^5)$ $O(\|E\|^5 \log\|E\|)$
Weighted Intersection of $p \geq 3$ Matroids [7]	$O(\|E\|^{p+2}(\log r(E) + \log W_{\max}))$	RLS_{p+1} (1+1) EA	$p+1$ $\|E\|$	$(p+2)^{\|E\|/2}$ $\|E\|^{\|E\|/2}$	$O(\|E\|^{p+3} \log p)$ $O(\|E\|^{p+3} \log\|E\|)$
Minimum Spanning Tree [5]	$O(\|E\|\|V\|(\log\|V\| + \log W_{\max}))$ (1st ph.) $O(\|E\|\|V\|^2)$ (2nd phase)	SEMO GSEMO	$\|E\|$ (!) $\|E\|$	$\|E\|^{\|E\|/2}$ $\|E\|^{\|E\|/2}$	$O(\|E\|^2\|V\| \log\|V\|)$ $O(\|E\|^2\|V\| \log\|V\|)$
Minimum Set Cover [2]	$O(\|S\|^2\|C\| + \|S\|\|C\|(\log\|C\| + \log W_{\max}))$	SEMO GSEMO	$\|C\|$ (!) $\|C\|$	$\|C\|^{\|C\|/2}$ $\|C\|^{\|C\|/2}$	$O(\|S\|^2\|C\| + \|S\|\|C\|^2 \log\|C\|)$ $O(\|S\|^2\|C\| + \|S\|\|C\|^2 \log\|C\|)$

Table 1. Application of the upper bound of Theorem 3 to known results depending on $W_{\max}$ (see Section 4 for a detailed discussion). The new results are obtained by replacing $W_{\max}$ by the upper bound on $w_{\max}$. Note that $n = |E|$ or $n = |C|$, respectively.Furthermore, in the Minimum Spanning Tree problem we have $\log|E| = O(\log|V|)$. The results for $\ell = 2$ are trivial and are already mentioned in [7]. The results for Weighted Matroid Intersection, Weighted Intersection of $p \geq 3$ Matroids and Minimum Set Cover correspond to $1/2$-, $1/p$- and $\log|S|$-approximate solutions, respectively.

replacement of the given weights W_i by the new weights w_i is only done conceptually. The randomized search heuristic still runs on the given weights W_i. Only the runtime analysis is based on the new weights w_i. If the new weights are chosen such that the behavior of RSH_ℓ does not change, then an upper bound on the optimal $w_{\max}$ can be used in the runtime analysis instead of $W_{\max}$.

Consider for example the weights $W = (3, 7, 11, 19, 31)$ and $\ell = 3$. These weights can be replaced by $w = (1, 2, 4, 7, 12)$, because $f_W(x) - f_W(x') = \sum_{i=1}^n W_i x_i - \sum_{i=1}^n W_i x'_i$ has the same sign as $f_w(x) - f_w(x') = \sum_{i=1}^n w_i x_i - \sum_{i=1}^n w_i x'_i$ for all $x, x' \in S$ with $H(x, x') \leq 3$. On the other hand, consider the weights $W = (3, 5, 7, 11, 17)$. In this case, there are no weights w with $w_{\max} < 17$ satisfying the conditions above.

A lower bound on the largest minimal $w_{\max}$ is interesting for worst-case analyses. Such a bound implies the existence of problem instances with weights of a certain size such that these weights cannot be replaced by smaller weights without affecting the behavior of the randomized search heuristic. The second example given above is such a worst-case instance for $n = 5$ and $\ell = 3$.

In this paper we show that for any given weights $W_1, \ldots, W_n$ there are always equivalent weights $w_1, \ldots, w_n$ such that $w_{\max} \leq n^{n/2}$. Two weight vectors are called equivalent if the behavior of RSH_ℓ does not change by replacing one weight vector with the other one in the objective function. Depending on ℓ this bound can be improved significantly, for example, for $\ell = 3$ we have $w_{\max} \leq \frac{1}{2}\sqrt{3} \cdot 2^n$. These results have important consequences for optimization problems where the runtime analysis of evolutionary algorithms depends on $W_{\max}$. We obtain the first strongly polynomial bounds for problems for which only weakly polynomial bounds were previously known. We summarize these results in Table 1 (see Section 4 for a detailed discussion).

The remainder of this work is structured as follows. In Section 2 we give a formal definition of the considered problem. The main results are proved in Section 3, where we show lower and upper bounds on the largest minimal $w_{\max}$. In Section 4 we apply these results to optimization problems where the runtime analyses of evolutionary algorithms depends on $W_{\max}$. Experimental results for $\ell = 3$ and $\ell = n$ are presented in Section 5. Finally we conclude our work in Section 6.

2. PROBLEM DEFINITION

Let $\mathrm{sign}(\cdot)$ denote the three-valued sign function

$$\mathrm{sign} : \mathbb{R} \mapsto \{-1, 0, 1\},\ \mathrm{sign}(y) = \begin{cases} +1, & y > 0 \\ 0, & y = 0 \\ -1, & y < 0 \end{cases}.$$

Furthermore, for $z \in \mathbb{R}^n$ let $|z|_{\neq 0}$ denote the number of entries not equal to zero.

The difference of the objective values of two search points $x \in F$ and $x' \in F$ can be written as

$$f(x') - f(x) = \sum_{i=1}^n W_i x'_i - \sum_{i=1}^n W_i x_i = \sum_{i=1}^n d_i W_i\,,$$

with $d := x' - x \in \{-1, 0, 1\}^n$. If $H(x, x') \leq \ell$, we have $|d|_{\neq 0} \leq \ell$. Hence our problem can be stated as follows.

Problem 1. (Weight Minimization Problem) *Given n weights $W_1, \ldots, W_n \in \mathbb{N}$, $0 < W_1 \leq W_2 \leq \ldots \leq W_n$, and $\ell \in \mathbb{N}$. Find weights $w_1, \ldots, w_n \in \mathbb{N}$, w_n minimal, such that $0 < w_1 \leq \ldots \leq w_n$ and*

$$\mathrm{sign}\left(\sum_{i=1}^n d_i w_i\right) = \mathrm{sign}\left(\sum_{i=1}^n d_i W_i\right) \tag{1}$$

for all $d \in \{-1, 0, 1\}^n, 2 \leq |d|_{\neq 0} \leq \ell$.

For simplicity, we require all weights to be sorted in non-decreasing order. Hence, W_n and w_n take the role of $W_{\max}$ and $w_{\max}$. Note that we explicitly allow non-unique weights, because non-unique weights $W_i = W_{i+1}$ can be used to encode constraints such as $W_k = W_i + W_{i+1} = 2W_i$. Also note that the conditions (1) for $|d|_{\neq 0} = 0$ and $|d|_{\neq 0} = 1$ are fulfilled trivially.

Algorithm 1 does not differentiate between $f(x') > f(x)$ and $f(x') = f(x)$, while the three-valued sign function does. This is intended, since x and x' might appear in the algorithm in interchanged roles. Hence, we have to distinguish all three cases.

Note that the conditions (1) are sufficient for our original motivation, but not always necessary. In particular, if $F \subset S$ there might be a $d \in \{-1,0,1\}^n$ such that there is no $x, x' \in F$ with $x - x' = d$. In this case, our formulation of the weight minimization problem contains conditions that are not necessary for $w_1, \ldots, w_n$ being equivalent to $W_1, \ldots, W_n$ and . In the following, we assume the worst case $F = S$, i. e., all constraints are necessary (in the sense that they do not impose additional restrictions, some of them are still redundant).

The right-hand sides of the conditions (1) are fixed numbers in $\{-1, 0, 1\}$. We divide these conditions into three classes based on their right-hand side. Let

$$LT := \left\{ d \in \{-1,0,1\}^n \;\middle|\; 2 \le |d|_{\neq 0} \le \ell, \sum_{i=1}^{n} d_i W_i \le -1 \right\},$$

$$EQ := \left\{ d \in \{-1,0,1\}^n \;\middle|\; 2 \le |d|_{\neq 0} \le \ell, \sum_{i=1}^{n} d_i W_i = 0 \right\}, \text{ and}$$

$$GT := \left\{ d \in \{-1,0,1\}^n \;\middle|\; 2 \le |d|_{\neq 0} \le \ell, \sum_{i=1}^{n} d_i W_i \ge 1 \right\}.$$

Since all d_i and W_i are integral, we have $LT \mathbin{\dot\cup} EQ \mathbin{\dot\cup} GT = \{d \in \{-1,0,1\}^n \mid 2 \le |d|_{\neq 0} \le \ell\}$. Using this notation we can restate Problem 1 as follows.

Problem 2. *Given n weights $W_1, \ldots, W_n \in \mathbb{N}$, $0 < W_1 \le W_2 \le \ldots \le W_n$, and $\ell \in \mathbb{N}$. Find weights $w_1, \ldots, w_n \in \mathbb{N}$, w_n minimal, such that $w_1 > 0$,*

$$\sum_{i=1}^{n} d_i w_i \le -1 \quad \text{for all } d \in LT,$$
$$\sum_{i=1}^{n} d_i w_i = 0 \quad \text{for all } d \in EQ, \text{ and}$$
$$\sum_{i=1}^{n} d_i w_i \ge 1 \quad \text{for all } d \in GT.$$

Note that all constraints with d lexicographically smaller than $(0, \ldots, 0)$ can be omitted from this description since they are implied by the corresponding constraint for $-d$.

We would like to mention the following geometric interpretation of Problem 2. The vector W can be interpreted as the normal of a hyperplane in $\mathbb{R}^n$ through the origin. This hyperplane partitions the set $\{d \in \{-1,0,1\}^n \mid 2 \le |d|_{\neq 0} \le \ell\}$ into three subsets corresponding to the points below, on, and above the hyperplane. The task is to find a hyperplane through the origin that maintains this partition and whose normal has integral components and minimal infinity norm.

Let $w_n^{\ell *} := w_n^{\ell *}(W_1, \ldots, W_n, \ell)$ denote the smallest w_n of all solutions to a given instance $(W_1, \ldots, W_n, \ell)$. Furthermore let $w_n^{\ell **} := \max_W w_n^{\ell *}(W_1, \ldots, W_n, \ell)$ denote the largest $w_n^{\ell *}$ over all instances for fixed parameters n and ℓ. We are interested in lower and upper bounds on $w_n^{\ell **}$. We use the upper index ℓ in $w_n^{\ell *}$ and ${}_n^{\ell **}$ to stress the dependence on ℓ. For simplicity, we drop this index in general discussions about the problem.

We remark that Problem 2 has a straightforward integer programming (IP) formulation with n variables and $1 + |LT| + |EQ| + |GT| \in O(\min\{n^\ell, 3^n\})$ constraints. For $\ell = 3$ there is a better formulation using only n^2 constraints (see Section 5.1) which can be easily solved by IP solvers, e. g., random instances up to $n = 1000$ can be solved within seconds. Our focus is not to develop a combinatorial algorithm to solve given instances of the problem. Rather we are interested in lower and upper bounds on the optimal w_n over all input weights W_i.

3. LOWER AND UPPER BOUNDS

The case $\ell = 2$ is trivial. The optimum weights w_i are given by $w_i = |\{W_1, \ldots, W_i\}|$. Hence, $w_i \le i$ and $w_n^{2*} \le n$. Considering the weights $W_i = i$, we obtain $w_n^{2**} = n$.

We assume $\ell \ge 3$ in the remainder of this section.

3.1 Lower Bounds

First, we give constructive lower bounds by considering specific inputs W_i such that $w_i = W_i$ is an optimal solution. Later, we prove a better, non-constructive lower bound for the case $\ell = n$. This bound can be generalized to $\ell < n$ but leads only to weak bounds in the general case.

Constructive lower bounds

The constructive lower bounds are based on Fibonacci numbers.

Proposition 1. *Let $n \in \mathbb{N}$, $n \ge 3$, $\epsilon > 0$ and $\phi = \frac{1}{2}(1 + \sqrt{5})$. Then $w_n^{3**} \ge \frac{1}{\sqrt{5}} \cdot \phi^{n+1} - \epsilon$ for all $n \ge n_0$ for some $n_0(\epsilon) \in \mathbb{N}$.*

Proof. Let F_i denote the i-th Fibonacci number (starting with $F_1 = F_2 = 1$) and define $W_i = F_{i+1}$. We have $W_{i-2} + W_{i-1} = F_{i-1} + F_i = F_{i+1} = W_i$ for all $i \in \mathbb{N}, i \ge 3$. Obviously, $w_i = W_i, i = 1, \ldots, n$ is the optimal solution to Problem 1. Thus, $w_n^{3**} \ge w_n^{3*} = W_n = F_{n+1}$.

Since $F_n = \frac{1}{\sqrt{5}}(\phi^n - (1-\phi)^n)$ and $\lim_{n\to\infty}(1-\phi)^n = 0$, there exists an n_0 (depending on ϵ) such that $w_n^{3**} \ge \frac{1}{\sqrt{5}} \cdot \phi^{n+1} - \epsilon$ for all $n \ge n_0$. $\square$

The bound in Proposition 1 also holds for $\ell > 3$, although we can improve this bound using generalized Fibonacci numbers. The Fibonacci k-step numbers $\left(F_i^{(k)}\right)_{i=1}^{\infty}$, $k \ge 2$ are defined as

$$F_i^{(k)} = 0 \quad \text{for all } i \le 0,$$
$$F_1^{(k)} = 1,$$
$$F_i^{(k)} = \sum_{j=1}^{k} F_{i-j}^{(k)} \quad \text{for all } i \ge 2.$$

The ratio $F_i^{(k)}/F_{i-1}^{(k)}$ converges to ϕ_k where ϕ_k is the positive root greater than 1 of $x^k - x^{k-1} \ldots - x - 1$. See Table 2 for the first values of ϕ_k. Note that $\phi_2 = \phi$. Subtracting the definition of $F_{i-1}^{(k)}$ from the definition of $F_i^{(k)}$ yields the three term recursion formula

$$F_i^{(k)} = 2F_{i-1}^{(k)} - F_{i-k-1}^{(k)} \quad \text{for all } i \ge 3.$$

Therefore, ϕ_k is bounded from above by 2.

k	ϕ_k (approx.)	name
2	1.618033989	Fibonacci constant
3	1.839286755	Tribonacci constant
4	1.927561975	Tetranacci constant
5	1.965948237	Pentanacci constant
6	1.983582843	Hexanacci constant
7	1.991964197	Heptanacci constant
8	1.996031180	Octanacci constant
9	1.998029470	Enneanacci constant
10	1.999018633	Decanacci constant

Table 2. Limit ϕ_k of the ratio of subsequent Fibonacci k-step numbers. The limit is given by the real root $\xi \geq 1$ of $x^k - x^{k-1} \ldots - x - 1$.

THEOREM 1. *Let $n \in \mathbb{N}$, $\ell \geq 3$, and $\epsilon > 0$. Then $w_n^{\ell **} \in \Omega((\phi_{\ell-1} - \epsilon)^n)$.*

PROOF. Define $W_i = F_{i+1}^{(\ell-1)}$ for $i \geq 1$. It holds

$$W_i = F_{i+1}^{(\ell-1)} = \sum_{j=1}^{\ell-1} F_{i+1-j}^{(\ell-1)} = \sum_{j=1}^{\ell-1} W_{i-j}$$

(assuming $W_i := 0$ for $i \leq 0$). Then $w_i = W_i$, $i = 1, \ldots, n$ is the optimal solution for the given weights W_i. Thus, $w_n^{\ell **} \geq w_n^{\ell *} = W_n = F_{n+1}^{(\ell-1)}$. Since $F_i^{(\ell-1)}/F_{i-1}^{(\ell-1)}$ converges to $\phi_{\ell-1}$, there is an n_0 such that $F_i^{(\ell-1)}/F_{i-1}^{(\ell-1)} \geq \phi_{\ell-1} - \epsilon$ for all $n \geq n_0$. □

For $\ell = 3$ the result of Proposition 1 can be improved by a constant factor of slightly less than ϕ as follows.

PROPOSITION 2. *Let $n \in \mathbb{N}$, $n \geq 3$, $\epsilon > 0$ and $\phi = \frac{1}{2}(1 + \sqrt{5})$. Then $w_n^{3**} \geq \frac{1}{\sqrt{5}} \cdot \phi^{n+2} - 1 - \epsilon$ for all $n \geq n_0(\epsilon)$.*

PROOF. Define $W_i = F_{i+2} - 1$. We have $W_{i-2} + W_{i-1} = F_i - 1 + F_{i+1} - 1 = F_{i+2} - 2 = W_i - 1 < W_i$ for all $i \in \mathbb{N}, i \geq 3$. Obviously, $w_i = W_i, i = 1, \ldots, n$ is the optimal solution to Problem 1, and hence, $w_n^{3**} \geq w_n^{3*} = F_{n+2} - 1$.

Since $F_n = \frac{1}{\sqrt{5}}(\phi^n - (1-\phi)^n)$ and $\lim_{n\to\infty}(1-\phi)^n = 0$, there exists an n_0 (depending on ϵ) such that $w_n^{3**} \geq \frac{1}{\sqrt{5}} \cdot \phi^{n+2} - 1 - \epsilon$ for all $n \geq n_0$. □

A similar construction leads to an explicit bound for $\ell = n$. Let $W_1 := 1$ and $W_i := 1 + \sum_{j=1}^{i-1} W_j$. Then $w_n^{n**} \geq w_n^{n*} = 2^{n-1}$.

Non-constructive lower bounds

For $\ell = n$ we can obtain a much better lower bound. ALON and VU [1] consider the problem of minimizing weights for threshold gates. A threshold gate is a function $f_n : \{-1,1\}^n \mapsto \{-1,1\}$ defined by $f_n(x_1, \ldots, x_n) = \mathrm{sign}(\sum_{i=1}^n W_i x_i - T)$, where the the weights $W_1, \ldots, W_n$ and the parameter T are chosen such that $\sum_{i=1}^n W_i x_i - T \neq 0$ for all $x \in \{-1,1\}$. It is easy to see that every threshold gate can be realized by integral weights W_i. A natural question is how large one has to choose these integral weights in the worst case.

Following the notation introduced in Section 2 we define

$$LT' := \left\{ d \in \{-1,1\}^n \,\middle|\, \sum_{i=1}^n d_i W_i - T \leq -1 \right\} \text{ and}$$

$$GT' := \left\{ d \in \{-1,1\}^n \,\middle|\, \sum_{i=1}^n d_i W_i - T \geq 1 \right\}.$$

Now the weight minimization problem for threshold gates can be stated as follows.

PROBLEM 3. *Given n weights $W_1, \ldots, W_n \in \mathbb{N}$, $T \in \mathbb{N}$. Find weights $w_1, \ldots, w_n \in \mathbb{N}$ and $t \in \mathbb{N}$ minimizing $\max\{w_1, \ldots, w_n\}$, such that*

$$\sum_{i=1}^n d_i w_i - t \leq -1 \qquad \text{for all } d \in LT', \text{ and}$$

$$\sum_{i=1}^n d_i w_i - t \geq 1 \qquad \text{for all } d \in GT'.$$

ALON and VU [1] prove the following result.

PROPOSITION 3. *Let $n \in \mathbb{N}$. There is a threshold gate f_n with $T = 0$ such that, if one restricts oneself to integral weights, the largest weight is at least*

$$\frac{n^{n/2}}{2^{n(2+o(1))}}.$$

Note that the property $T = 0$ is not explicitly spelled out in [1, Theorem 3.3.1], but the proof constructs a threshold gate such that $T = 0$. For n being a power of 2 an explicit bound of

$$\frac{1}{n} e^{-4n^\beta} \cdot \frac{n^{n/2}}{2^n}$$

where $\beta = \log(3/2)$ can be found in [3]. Using the result of ALON and VU we can prove the same lower bound for our problem.

THEOREM 2. *Let $n \in \mathbb{N}$. Then*

$$w_n^{n**} \geq \frac{n^{n/2}}{2^{n(2+o(1))}}.$$

PROOF. Let B denote the bound in the theorem. By Proposition 3 there is a threshold gate f_n with $T = 0$ such that the largest weight is at least B. Consider the corresponding weight vector $W = (W_1, \ldots, W_n)$. By symmetry of threshold gates, we can assume that $0 \leq W_1 \leq \ldots \leq W_n$. Consider the case $W_1 > 0$ first. Consider W as input to Problem 2 and assume that $w_n^{n**} < B$. Then there is a solution w to Problem 2 such that $w_n < B$. However, the weights w are also a solution to Problem 3, contradicting the choice of f_n. Hence, $w_n^{n**} \geq B$.

If $W_1 = 0$, let $r := \max\{i \mid W_i = 0\}$ and $n' := n - r$. Consider the vector $W' = (W_{r+1}, \ldots, W_n)$. Using the same argument as above for W' instead of W, we get $w_{n'}^{n'**} \geq B$, which gives an even stronger bound (for n') than claimed. In particular, there is a solution $w' = (w'_1, \ldots, w'_{n'})$ to Problem 2 for input W' with $w'_{n'}$ minimal and $w'_{n'} \geq B$. Now obtain the weights W'' by augmenting w' by r copies of $w'_{n'}$ and consider W'' again as input to Problem 2. Obviously, we have ${w''}_n^* \geq w'_{n'}$, since otherwise this would contradict the minimality of $w'_{n'}$. Thus, we have $w_n^{n**} \geq {w''}_n^* \geq w'_{n'} \geq B$. □

The result of Theorem 2 can be used to derive a similar, but weaker result for $\ell < n$. Solving Problem 2 for any subset of cardinality ℓ from the input weights yields a natural lower bound for the original problem.

COROLLARY 1. *Let $n \in \mathbb{N}$ and $\ell \leq n$. Then*

$$w_n^{\ell **} \geq \frac{\ell^{\ell/2}}{2^{\ell(2+o(1))}}.$$

However, in light of Theorem 1 this result is only useful for values ℓ close to n.

3.2 Upper Bound

To derive an upper bound on $w_n^{\ell**}$ we need an upper bound on the determinant of a matrix. Such a bound can be obtained from Hadamard's inequality.

PROPOSITION 4. *Let $A \in \{-1,0,1\}^{n\times n}$ with at most ℓ non-zero entries per row. Then $|det(A)| \leq \ell^{n/2}$. If A has at least one row with at most $\ell-1$ non-zeroes, then $|det(A)| \leq \sqrt{(\ell-1)/\ell}\cdot \ell^{n/2}$.*

PROOF. By Hadamard's inequality we have

$$|det(A)| \leq \prod_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}^2\right)^{1/2} \leq \prod_{i=1}^{n}\sqrt{\ell} = \ell^{n/2}.$$

The second results follows since at least one of the n factors $\sqrt{\ell}$ can be replaced with $\sqrt{\ell-1}$. $\square$

Now we are able to prove an upper bound on $w_n^{\ell**}$.

THEOREM 3. *Let $n \in \mathbb{N}$, $\ell \in \mathbb{N}$ and $\ell \leq n$. Then $w_n^{\ell**} \leq \sqrt{\ell/(\ell+1)}\cdot(\ell+1)^{n/2}$ holds. Furthermore, $w_n^{n**} \leq n^{n/2}$.*

PROOF. We prove that any optimal solution $w_n^{\ell*}$ of a given instance of Problem 2 is bounded as claimed. Then $w_n^{\ell**}$ is bounded in the same way.

Consider the natural IP formulation of Problem 2. This IP is feasible since $w_i = W_i$ is a feasible solution. Let x denote a basic feasible solution of the linear relaxation. There exist n linearly independent constraints satisfied with equality. Hence, we have $Ax = b$, where the rows of $A \in \{-1,0,1\}^{n\times n}$ are linearly independent, and $b \in \{-1,0,1\}^n$.

By Cramer's rule we have $x_i = det(A)^{-1}\cdot det(A_i|b) \geq 0$, where $A_i|b$ denotes matrix A with the i-th column replaced by b. Define $x_i' := |det(A)|\cdot x_i \geq 0$.

Note that A has at most ℓ non-zeroes per row, hence, $A_i|b$ has at most $\ell+1$ non-zeroes per row. Since A is non-singular, there is at least one non-zero entry in the i-th column of A. Hence, $A_i|b$ has at least one row with at most ℓ non-zeroes. By Proposition 4, we have

$$x_i' = |det(A)|\cdot x_i = |det(A_i|b)| \leq \sqrt{\ell/(\ell+1)}\cdot(\ell+1)^{n/2}.$$

The components of x_i' are determinants of a matrix with entries in $\{-1,0,1\}$, and hence, x' is integral. It can easily be verified that $x' \in \mathbb{Z}^n$ is a feasible solution. Since $w_n^{\ell*}$ is optimal, it is not larger than w_n of any solution, and hence, $w_n^{\ell*} \leq x_n' \leq \sqrt{\ell/(\ell+1)}\cdot(\ell+1)^{n/2}$.

If $\ell = n$, then $A_i|b$ has at most n non-zeroes per row and the claimed results follows. $\square$

Note that for $\ell = n$ the gap between the lower bound in Theorem 2 and the upper bound in Theorem 3 is $2^{n(2+o(1))}$. An interesting open problem is to close this gap.

4. APPLICATIONS

An immediate consequence of the lower bound of Theorem 1 is that there are instances of Problem 1 with $W_n \in \Omega((\phi_{\ell-1}-\epsilon)^n)$ such that the weights cannot be replaced by smaller weights without affecting the set of accepted transitions from x to x' in Algorithm 1. Examples of such worst-case instances for $\ell = 3$ are given in Section 5.2. Due to the lower bound in Theorem 2 we know that for $\ell = n$ there exist worst-case instances with

$$W_n \geq \frac{n^{n/2}}{2^{n(2+o(1))}}.$$

In particular, there is no fixed $a > 1$ such that $w_n^{n**} \in O(a^n)$.

The application of the upper bound in Theorem 3 to known results with runtimes depending on the largest weight is summarized in Table 1. The table presents several combinatorial optimization problems for which the performance of evolutionary algorithms has been analyzed. In the minimum spanning tree problem $|V|$ and $|E|$ denote the number of vertices and edges, respectively. In the matroid problems $|E|$ and $r(E)$ denote the size of the ground set E and the rank of the matroid, respectively. Note that $n = |E|$ in all cases. In the minimum set cover problem $|S|$ and $|C|$ denote the size of the ground set and the number of subsets, respectively. In this case, $n = |C|$.

First we focus on the results for two evolutionary algorithms called RLS and (1+1) EA. The (1+1) EA obtains a new search point x' by flipping the bits of a given search point x uniformly at random with probability $1/n$. The RLS algorithm picks one or two bits to be flipped according to a fixed probability distribution. Its variant RLS$_\ell$ picks up to ℓ such bits. The objective function used in the studies of the considered problems is linear in the weights (if restricted to feasible solutions in F). Hence, our results for Problem 1 can be transferred back to the original problem.

The RLS algorithm itself leads to the trivial case $\ell = 2$ which was already mentioned in [7]. Its variant RLS$_3$ used in the Weighted Matroid Intersection problem was the original motivation for this study (see also the experimental results for this special case in Section 5). While the number of bit flips in the RLS algorithm is bounded by a small number, the (1+1) EA algorithm might flip all bits of a search point in one iteration (although the probability of this event is exponentially small). Therefore, it is necessary to choose ℓ equal to $n = |E|$. This leads to worse bounds for (1+1) EA compared to RLS and its variants.

The last two examples in Table 1 take a special position since the SEMO and GSEMO algorithms do not fit into our framework of randomized search heuristics presented in the introduction. The SEMO and GSEMO algorithms are generalizations of RLS and (1+1) EA that maintain a set of search points called population. A newly generated search point x' is not only compared to its predecessor x, but to all search points in the population. Hence, if we choose ℓ equal to $|E|$ or $|C|$, respectively (even though SEMO flips only at most one bit per iteration), our results can also be applied to this case.

We remark that there are other problems where the runtime analysis of randomized search heuristics depends on the largest weight. However, our approach cannot be applied to these problems. For example, using the DEMO algorithm with $\epsilon = \Theta(1/m)$ the expected number of iterations to solve the minimum s-t-cut problem is $O(|E|^3(\log^2|V| + \log^2 W_{\max}))$ [4]. Unfortunately, the used objective function is not a linear function as introduced in Section 2, since it involves the value of a maximum s-t-flow. Moreover, the diversity mechanism used by DEMO is not invariant under weight changes as considered in this paper.

5. EXPERIMENTAL RESULTS

In this section we present some experimental results for the cases $\ell = 3$ and $\ell = n$. The case $\ell = 3$ is the smallest value for ℓ for which the problem is non-trivial. Furthermore, it has a special structure that admits an improved IP formulation and it is of interest for the largest common independent set in two matroids [7]. The case $\ell = n$ considers the largest possible value for ℓ. This case occurs for example in evolutionary algorithms such as (1+1) EA, SEMO and GSEMO, where search points of arbitrary large Hamming distances are compared to each other.

5.1 Improved IP Formulation for $\ell = 3$

In this section we consider the special case $\ell = 3$. Problem 2 can be formulated as an IP in the following way.

$$\begin{aligned} \text{minimize } & w_n \\ \text{s.t. } & w_1 \geq 1 \\ & \sum_{i=1}^{n} d_i w_i \leq -1 \quad \text{for all } d \in LT \\ & \sum_{i=1}^{n} d_i w_i = 0 \quad \text{for all } d \in EQ \\ & \sum_{i=1}^{n} d_i w_i \geq 1 \quad \text{for all } d \in GT \\ & w_i \in \mathbb{Z} \quad \text{for all } 1 \leq i \leq n \end{aligned} \tag{2}$$

We are interested in worst case instances, i. e., instances such that $w_n^* = w_n^{**}$. To obtain such instances one could enumerate all partitions $LT \mathbin{\dot\cup} EQ \mathbin{\dot\cup} GT$ of $\{d \in \{-1,0,1\}^n \mid 2 \leq |d|_{\neq 0} \leq 3\}$ and solve the corresponding IP. This approach is very inefficient since a large fraction of such partitions implies an infeasible IP. And if the IP is feasible, many constraints are redundant. Therefore we use another, more efficient IP formulation.

In the improved IP formulation the partition $LT \mathbin{\dot\cup} EQ \mathbin{\dot\cup} GT$ is replaced by a vector and an upper right triangular matrix. Let $b \in \{0,1\}^{n-1}$ denote a vector and $A = (a_{j,k})_{j,k} \in \{0,\ldots,2n\}^{n\times n}$ an upper right triangular matrix. The integer program $\mathrm{IP}(A,b)$ corresponding to the matrix A and vector b is defined as

$$\begin{aligned} \text{minimize } & w_n \\ \text{s.t. } & w_1 \geq 1 \\ w_i - w_{i-1} \geq 1 \quad & \text{for all } 2 \leq i \leq n, b_{i-1} = 1 \\ w_i - w_{i-1} = 0 \quad & \text{for all } 2 \leq i \leq n, b_{i-1} = 0 \\ w_j + w_k - w_{a_{j,k}/2+1} \leq -1 \quad & \text{for all } 1 \leq j < k \leq n, a_{j,k} \text{ even,} \\ & a_{j,k}/2 + 1 \leq n \\ w_j + w_k - w_{(a_{j,k}+1)/2} = 0 \quad & \text{for all } 1 \leq j < k \leq n, a_{j,k} \text{ odd} \\ w_j + w_k - w_{a_{j,k}/2} \geq 1 \quad & \text{for all } 1 \leq j < k \leq n, a_{j,k} \text{ even,} \\ & a_{j,k}/2 \geq 1 \\ w_i \in \mathbb{Z} \quad & \text{for all } 1 \leq i \leq n \end{aligned}$$

The vector component b_{i-1} encodes whether $w_i = w_{i-1}$ or $w_i > w_{i-1}$ should hold. The matrix entry $a_{j,k}$ encodes conditions for the range of the sum $w_j + w_k$. If $a_{j,k}$ is odd, then $w_j + w_k$ equals weight w_i where $i = (a_{j,k}+1)/2$. If $a_{j,k}$ is even, $w_i + 1 \leq w_j + w_k \leq w_{i+1} - 1$ holds where $i = a_{j,k}/2$ and $w_0 := 0$, $w_{n+1} := \infty$.

n	# △ matr.	# enum. △ matr.	# feas. IPs
1	1	1	1
2	1	1	2
3	3	3	8
4	125	22	46
5	$1.2 \cdot 10^5$	372	442
6	$3.4 \cdot 10^9$	10.936	6.395
7	$4.2 \cdot 10^{15}$	479.064	131.711
8	$2.4 \cdot 10^{23}$	30.846.418	3.658.432
9	$8.5 \cdot 10^{32}$	2.953.407.869	130.833.291
10	$2.0 \cdot 10^{44}$	433.550.516.563	5.822.596.188

Table 3. Total number of triangular matrices, number of enumerated triangular matrices and number of feasible IPs.

Given weights $W_1,\ldots,W_n$ it is straightforward to compute the matrix A and vector b such that $w \in \mathbb{N}^n$ is a solution to $\mathrm{IP}(A,b)$ if and only if w is a solution to Problem 1. Likewise, given a partition $LT \mathbin{\dot\cup} EQ \mathbin{\dot\cup} GT$ of $\{d \in \{-1,0,1\}^n \mid 2 \leq |d|_{\neq 0} \leq 3\}$ such that the corresponding IP is feasible, one can easily compute the matrix A and vector b such that both IPs have the same set of solutions. The reverse transformation is also straightforward for matrices A and vectors b such that $\mathrm{IP}(A,b)$ is feasible.

The new formulation has at most n^2 constraints. We can easily derive necessary conditions on A such that there exists a vector b such that $\mathrm{IP}(A,b)$ is feasible. By monotonicity of w_i we have $w_j + w_k \geq w_1 + w_2 > w_2$, and hence

$$a_{j,k} \geq 4 \qquad \text{for all } 1 \leq j < k \leq n, \tag{3}$$

that is, all matrix entries are restricted to $\{4,\ldots,2n\}$. We have $w_j + w_n > w_n$, which implies

$$a_{j,n} = 2n \qquad \text{for all } 1 \leq j < n, \tag{4}$$

that is, the last column of A is fixed to $2n$. More generally, we have $w_j + w_k > w_k$, and hence

$$a_{j,k} \geq 2k \qquad \text{for all } 1 \leq j < k \leq n. \tag{5}$$

The monotonicity of w_i carries over to $a_{j,k}$: We have $w_j + w_k \geq w_{j-1} + w_k$ and $w_j + w_k \geq w_j + w_{k-1}$. This implies

$$a_{j,k} \geq a_{j-1,k} \qquad \text{for all } 1 < j < k \leq n, \tag{6}$$

$$a_{j,k} \geq a_{j,k-1} \qquad \text{for all } 1 \leq j < k < n. \tag{7}$$

The set of upper right triangular matrices satisfying (3), (4), (5), (6) and (7) can be easily enumerated. The columns of A can be interpreted as a vector of dimension $n(n-1)/2$ with entries in $\{4,\ldots,2n\}$. Due to equation (4) we can ignore the last column of A and reduce the dimension of the vector to $(n-1)(n-2)/2$. This vector can be interpreted as a number with $(n-1)(n-2)/2$ digits in a number system with base $2n+1$. By counting from 0 to $(2n+1)^{(n-1)(n-2)/2}-1$ we enumerate all upper right triangular matrices in $\{0,\ldots,2n\}^{n\times n}$ satisfying equation (4). The conditions (3), (5), (6) and (7) can be easily integrated in the enumeration process. Note that these conditions on the matrix A are necessary for the existence of some vector b such that $\mathrm{IP}(A,b)$ is feasible, but the conditions are not sufficient.

n	$W_1, \ldots, W_n$
1	1
2	1, 2
3	1, 2, 4 2, 3, 4
4	2, 3, 4, 8 2, 4, 5, 8 3, 4, 6, 8
5	3, 5, 7, 11, 17
6	4, 5, 10, 13, 16, 30
7	5, 17, 21, 25, 31, 37, 55
8	5, 17, 21, 25, 31, 37, 55, 93 7, 11, 19, 25, 31, 41, 51, 93
9	15, 25, 39, 53, 65, 69, 85, 91, 155
10	11, 49, 61, 73, 83, 93, 109, 157, 175, 267
11	11, 49, 61, 73, 83, 93, 109, 157, 175, 267, 443
12	11, 21, 33, 45, 55, 75, 101, 147, 249, 323, 397, 721

Table 4. Worst-case instances that maximize w_n^{3*} for $n \leq 12$. Values for $n = 11$ and $n = 12$ subject to the conjecture that there are no equality constraints in worst-case instances. Values for $n = 12$ conjectured to be a worst-case instance.

5.2 Results for $\ell = 3$

In Table 3 we present some numbers concerning the complexity of our approach. The total number of considered triangular matrices is $(2n-3)^{(n-1)(n-2)/2}$, i. e., conditions (3) and (4) are already taken into account here. With a little bit of extra work it is possible to skip matrices that do not satisfy conditions (5), (6) or (7). Hence, the number of actually enumerated triangular matrices is much smaller. The last column depicts the number of feasible integer programs $\mathrm{IP}(A, b)$. While some of the enumerated matrices lead to infeasible integer programs for any vector b, there are also matrices such that there are several vectors b where $\mathrm{IP}(A, b)$ is feasible.

A compilation of worst-case instances for $n \leq 12$ is given in Table 4. An instance is called worst-case instance if $W_n = w_n^{3*} = w_n^{3**}$. Note that there are two such instances for $n = 3$, 8, and 12. For $n = 4$ there are even three such instances. We remark that in these cases each instance corresponds to a different matrix/vector pair (A, b) and integer program $\mathrm{IP}(A, b)$. For all given worst-case instances there is exactly one optimal solution to the corresponding integer program $\mathrm{IP}(A, b)$.

In Table 5 we compare the lower bound from Proposition 2, the upper bound from Theorem 3, and the observed optimal values w_n^{3**} for $n \leq 12$.

5.3 Conjectures

We present two conjectures originating from the experimental results. By Theorem 3 we have $w_n^{3**} \in O(2^n)$. However, in the experimental results in Table 5 the ratio w_n^{3**}/w_{n-1}^{3**} approaches ϕ.

Conjecture 1. *For $n \in \mathbb{N}$ holds $w_n^{3**} \in O(\phi^n)$.*

Note that this upper bound is of the same order as the lower bound in Proposition 1, Theorem 1 and Proposition 2.

As can be seen in Table 4 there are no equality constraints in worst-case instances for $\ell = 3$.

n	low. bd.	w_n^{3**}	upp. bd.
1	1	1	1
2	2	2	3
3	4	4	6
4	7	8	13
5	12	17	27
6	20	30	55
7	33	55	110
8	54	93	221
9	88	155	443
10	143	267	886
11	232	≥ 443	1773
12	376	≥ 721	3547

Table 5. Lower and upper bounds on w_n^{3**}. The lower bound is $\lceil 1/\sqrt{5} \cdot \phi^{n+2} - 3/2 \rceil$, the upper bound is $\lfloor 1/2\sqrt{3} \cdot 2^n \rfloor$.

n	$W_1, \ldots, W_n$	
1	1	
2	1, 2	
3	1, 2, 4	2, 3, 4
4	2, 3, 4, 10 2, 3, 6, 10 2, 4, 7, 10	2, 6, 7, 10 3, 4, 8, 10 3, 6, 8, 10
5	4, 6, 11, 14, 30 4, 6, 11, 16, 30 4, 6, 14, 19, 30 4, 6, 16, 19, 30 4, 11, 14, 24, 30 4, 11, 16, 24, 30 5, 8, 12, 14, 30	5, 8, 12, 16, 30 5, 8, 14, 18, 30 5, 8, 16, 18, 30 5, 12, 14, 22, 30 5, 12, 16, 22, 30 6, 11, 14, 26, 30 8, 14, 18, 25, 30
6	10, 22, 27, 36, 40, 114 10, 22, 27, 36, 74, 114 10, 22, 27, 40, 78, 114	10, 22, 27, 74, 78, 114 10, 22, 36, 40, 87, 114 10, 27, 36, 40, 92, 114

Table 6. Worst-case instances that maximize w_n^{n*} for $n \leq 6$.

Conjecture 2. *Let $n \in \mathbb{N}$ and let $W_1, \ldots, W_n$ denote weights such that $w_n^{3*} = w_n^{3**}$. Then for all $i, j, k \in \{1, \ldots, n\}$, $j < k < i$ holds $W_j \neq W_k$ and $W_j + W_k \neq W_i$.*

5.4 Results for $\ell = n$

In the case $\ell = n$ the problem does not exhibit such a nice structure as for $\ell = 3$. Therefore, we perform an exhaustive search using the IP formulation (2). Note that the number of constraints can be reduced from $\frac{1}{2}(3^n + 1)$ to $2^n - 1$ as follows. Sort the search points $x \in \{0,1\}^n$ by their weight $f(x)$. Consider only those constraints in (2) where d is the difference of two adjacent search points in the sorted sequence. Since $\ell = n$ this subset of constraints implies all other constraints.

The results of such an exhaustive search among all nondecreasing vectors in $\{1, \ldots, \lfloor n^{n/2} \rfloor\}^n$ are shown in Table 6. For $7 \leq n \leq 13$ a compilation of bad instances, i. e., with large w_n^{n*}, is given in Table 7. In Table 8 we compare the known lower and upper bounds with the largest values for w_n^{n*} observed in our experiments. Here we used 2^{n-1} as lower bound (see remark after Proposition 2), although Theorem 2 yields an asymptotically better lower bound, but

n	$W_1, \ldots, W_n$
7	15, 32, 67, 72, 95, 146, 267
8	6, 19, 54, 104, 138, 538, 606, 822
9	95, 241, 262, 350, 682, 757, 844, 1993, 2078
10	82, 107, 160, 730, 766, 918, 1006, 3200, 3212, 6038
11	42, 872, 1993, 2966, 3095, 7159, 7165, 9751, 11880, 16010, 22024
12	2870, 7890, 11712, 14726, 15798, 17027, 18516 21457, 33910, 37217, 40292, 47266
13	5903, 10164, 24271, 24429, 30922, 31515, 59366, 69547, 74714, 87071, 98394, 105613, 123889

Table 7. Bad (but probably not worst-case) instances for $7 \leq n \leq 13$. These instances have largest w_n^{n*} among 10^5 instances with weights randomly chosen from the interval $[1, 10^6]$ and sorted.

n	low. bd.	w_n^{n**}	upp. bd.
1	1	1	1
2	2	2	2
3	4	4	5
4	8	10	16
5	16	30	55
6	32	114	216
7	64	≥ 267	907
8	128	≥ 822	4096
9	256	≥ 2078	19683
10	512	≥ 6038	100000
11	1024	≥ 22024	534145
12	2048	≥ 47266	2985984
13	4096	≥ 123889	17403307

Table 8. Lower and upper bounds on w_n^{n**}. The lower bound is 2^{n-1}, the upper bound is $\lfloor n^{n/2} \rfloor$. For $n \geq 7$ the middle column represents the largest w_n^{n*} found in 10^5 instances with weights randomly chosen from the interval $[1, 10^6]$ and sorted. The value given for w_6^{6**} is a conjecture.

more knowledge about the $o(1)$ term is required to obtain an explicit value. The first power of 2 for which the explicit bound in [3] gives a lower bound better than 2^{n-1} is $n = 128$.

6. CONCLUSIONS

We have analyzed the influence of the size of weights on the behavior of a certain class of randomized search heuristics. It turns out that it is not necessary to handle arbitrarily large weights. Instead it is possible to consider equivalent weights where the largest weight is bounded exponentially in the problem size.

This result allows to remove the dependency on $W_{\max}$ in the runtime analyses that have been carried out for evolutionary algorithms on several combinatorial optimization problems. In particular we obtain strongly instead of weakly polynomial bounds on the runtime of these algorithms. Additionally we give constructive as well as non-constructive lower bounds for the largest weight in worst-case instances. Finally we present experimental results for the important subclasses $\ell = 3$ and $\ell = n$ of the problem, including worst-case instances.

An open problem is to close the gap between the lower and the upper bounds. To this end it is probably helpful to understand the structure of worst-case instances. For the case $\ell = 3$ we state conjectures about a smaller upper bound and the structure of such worst-case instances.

Acknowledgements

We thank Friedrich Eisenbrand for pointing us to reference [1]. We also thank Carsten Witt for helpful discussions.

References

[1] N. Alon and V. H. Vu. Anti-Hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs. *Journal of Combinatorial Theory, Series A*, 79(1):133–160, 1997.

[2] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective models. In *Proc. of the 9th Genetic and Evolutionary Computation Conference (GECCO '07), London*, pages 797–804, 2007.

[3] J. Hastad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.

[4] F. Neumann, J. Reichel, and M. Skutella. Computing minimum cuts by randomized search heuristics. In *Proc. of the 10th Genetic and Evolutionary Computation Conference (GECCO '08), Atlanta, USA*, pages 779–786, 2008.

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

[6] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms and the minimum spanning tree problem. *Theoretical Computer Science*, 378(1):32–40, 2007.

[7] J. Reichel and M. Skutella. Evolutionary algorithms and matroid optimization problems. In *Proc. of the 9th Genetic and Evolutionary Computation Conference (GECCO '07), London*, pages 947–954, 2007.

[8] I. Wegener. Towards a theory of randomized search heuristics. In *Proc. of Mathematical Foundations of Computer Science (MFCS '03), Bratislava, Slovakia*, pages 125–141, 2003.

[9] I. Wegener. Randomized search heuristics as an alternative to exact optimization. In W. Lenski, editor, *Logic versus Approximation*, pages 138–149. 2004.