



**HAL**  
open science

# The First Proven Performance Guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a Combinatorial Optimization Problem

Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Yakob Kahane, Simon  
Wietheger

► **To cite this version:**

Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Yakob Kahane, Simon Wietheger. The First Proven Performance Guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a Combinatorial Optimization Problem. THE 32nd INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE IJCAI 2023, Aug 2023, Macao, Macau SAR China. pp.5522-5530, 10.24963/ijcai.2023/613 . hal-04103630

**HAL Id: hal-04103630**

**<https://hal.science/hal-04103630>**

Submitted on 23 May 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The First Proven Performance Guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a Combinatorial Optimization Problem\*

Sacha Cerf<sup>1</sup>, Benjamin Doerr<sup>2</sup>, Benjamin Hebras<sup>1</sup>,  
Yakob Kahane<sup>1</sup>, Simon Wietheger<sup>3</sup>

<sup>1</sup>École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France

<sup>2</sup>Laboratoire d’Informatique (LIX), CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France

<sup>3</sup>Hasso Plattner Institute, University of Potsdam, Germany<sup>†</sup>  
firstname.lastname@polytechnique.edu, simon.wietheger@student.hpi.de

## Abstract

The Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is one of the most prominent algorithms to solve multi-objective optimization problems. Recently, the first mathematical runtime guarantees have been obtained for this algorithm, however only for synthetic benchmark problems.

In this work, we give the first proven performance guarantees for a classic optimization problem, the NP-complete bi-objective minimum spanning tree problem. More specifically, we show that the NSGA-II with population size  $N \geq 4((n - 1)w_{\max} + 1)$  computes all extremal points of the Pareto front in an expected number of  $O(m^2nw_{\max} \log(nw_{\max}))$  iterations, where  $n$  is the number of vertices,  $m$  the number of edges, and  $w_{\max}$  is the maximum edge weight in the problem instance. This result confirms, via mathematical means, the good performance of the NSGA-II observed empirically. It also shows that mathematical analyses of this algorithm are not only possible for synthetic benchmark problems, but also for more complex combinatorial optimization problems.

As a side result, we also obtain a new analysis of the performance of the global SEMO algorithm on the bi-objective minimum spanning tree problem, which improves the previous best result by a factor of  $|F|$ , the number of extremal points of the Pareto front, a set that can be as large as  $nw_{\max}$ . The main reason for this improvement is our observation that both multi-objective evolutionary algorithms find the different extremal points in parallel rather than sequentially, as assumed in the previous proofs.

## 1 Introduction

Many optimization problems consist of several conflicting objectives. In such a situation, it is not possible to compute a single optimal solution. The most common solution concept therefore is to compute a set of Pareto optima (solutions which cannot be improved in one objective without accepting a worsening in another objective) and let a decision maker select the final solution based on their preference.

Besides mathematical programming approaches, evolutionary algorithms (EAs) are the standard approach to multi-objective problems with many successful applications [Zhou *et al.*, 2011]. EAs profit here from their general ability to work with sets of solutions (“populations”). The by far most prominent multi-objective evolutionary algorithm (MOEA) is the *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) proposed by Deb, Pratap, Agarwal, and Meyarivan [Deb *et al.*, 2002] (with over 50,000 citations on Google Scholar).

While very successful in practice, this algorithm is only little understood from a fundamental perspective, giving the users little general advice on how to optimally employ this algorithm, e.g., how to set its parameters right. In fact, it was only at AAAI 2022 that the first mathematical runtime analysis of the NSGA-II was presented [Zheng *et al.*, 2022], a work that was quickly followed up more runtime analyses on this algorithm (see the previous works section). All these works analyze the performance of the NSGA-II on simple benchmark problems, mostly multi-objective variants of the ONEMAX, LEADINGONES, and JUMP benchmarks well-studied in the theory of single-objective randomized search heuristics [Auger and Doerr, 2011].

In this work, we conduct the first mathematical runtime analysis of the NSGA-II on a classic combinatorial problem, namely the NP-complete bi-objective minimum spanning tree problem. In this problem, we are given an undirected graph with  $n$  vertices and  $m$  edges. In the basic single-objective version of the minimum spanning tree (MST) problem, we are also given non-negative integral edge weights and the task is to compute a minimum

---

\* Author-generated version of a paper appearing in the proceedings of IJCAI 2023.

<sup>†</sup>Work done while visiting École Polytechnique, France.

spanning tree. This problem is easily solved by classic algorithms. It has also been used to understand how EAs solve combinatorial optimization problems. As a first result in this direction, Neumann and Wegener [Neumann and Wegener, 2007] showed that the  $(1 + 1)$  EA computes a minimum spanning tree in an expected number of  $O(m^2 \log(nw_{\max}))$  iterations (and fitness evaluations). Here  $w_{\max}$  denotes the maximum edge weight. Using a balanced mutation operator, this can be improved to  $O(mn \log(nw_{\max}))$ .

In the bi-objective variant of the problem, we are given two weight functions and the target is to compute the Pareto front of the problem of computing a spanning tree minimizing both weight functions. This problem is NP-complete, but it is possible to compute in polynomial time the extremal points of the Pareto front [Hamacher and Ruhe, 1994]. The first result on how MOEAs solve this problem is [Neumann, 2007]. It shows that the global SEMO algorithm, a multi-objective analogue of the basic  $(1 + 1)$  EA, computes the extremal points of the Pareto front in an expected number of  $O(m^2 nw_{\min} (|F| + \log(nw_{\max})))$  iterations, where  $F$  denotes the set of extremal points of the Pareto front and  $w_{\min}$  denotes the minimum of the maximum edge weights of the two weight functions. As in [Neumann and Wegener, 2007], this guarantee improves by a factor of  $\Omega(m/n)$  when using balanced mutation.

We note that [Neumann and Witt, 2022] propose the convex global SEMO algorithm and show that it can solve the bi-objective MST problem in polynomial time regardless of  $w_{\max}$ . Given that this algorithm is very new and not yet established, we do not follow this line of research. [Neumann and Witt, 2022, Theorem 4] also implies a runtime bound for the classic global SEMO algorithm, but this becomes superior to ours only when the size of the Pareto front is at most  $nw_{\min}/\ell$ , where  $\ell$  is a problem parameter which in general can only be estimated by  $\Theta(m^2)$ . So this result appears to be an improvement only in special cases.

In this work, we conduct a mathematical runtime analysis of the NSGA-II on the bi-objective MST problem. This is the first runtime analysis of this algorithm on a combinatorial problem. Besides showing that such analyses are possible, it proves that also the NSGA-II can efficiently compute the extremal points of the Pareto front, and this in an expected number of  $O(m^2 \log(nw_{\max}))$  iterations, hence a number of  $O(Nm^2 \log(nw_{\max}))$  fitness evaluations, when the population size  $N$  is at least  $N \geq 4nw_{\min}$ . This results holds for various ways of generating the offspring population including the use of crossover. As in the previous works, we obtain a bound lower by a factor of  $\Omega(m/n)$  when using balanced mutation.

We note that our runtime guarantees are smaller than the ones proven in [Neumann, 2007] for the global SEMO by essentially a factor of  $|F|$ , the number of extremal points, for which the only general upper bound is  $nw_{\min}$ . This improvement stems from our observation that the NSGA-II makes progress towards the different extremal

points in parallel, whereas the proof in [Neumann, 2007] assumed that these were found sequentially. We show the same improvement for the GSEMO, lowering Neumann’s bound to  $O(m^2 nw_{\min} \log(nw_{\max}))$ .

## 2 Previous Work

The multi-objective version of the minimum spanning tree problem, usually called multi-criteria minimum spanning tree problem (mc-MST), is an important combinatorial optimization problem with many applications in network design. We refer to Ehrgott [Ehrgott, 2005] for an extensive discussion of the problem and the different algorithmic approaches to it. Being NP-complete, many heuristic approaches have been developed [Arroyo *et al.*, 2008], including many based on evolutionary algorithms [Knowles and Corne, 2000; Knowles and Corne, 2001; Bossek and Grimme, 2017; Parraga-Alava *et al.*, 2017; Majumder *et al.*, 2020].

In this work, we investigate how the NSGA-II solves the bi-objective MST problem. As in most theoretical works on randomized search heuristics, our aim is not so much finding the best possible algorithm to solve this problem (for this problem-specific algorithms will usually be superior), but we aim at understanding how a certain algorithm, here the NSGA-II, solves a certain problem. The broader aim is to understand which algorithms are suitable for which problems, what are the right parameter settings, and to detect possible shortcomings and remedies for these. We refer to the textbooks [Auger and Doerr, 2011] for a broader introduction into the research field of mathematical analyses of randomized search heuristics and its achievements.

We briefly review the most relevant literature. The mathematical runtime analysis of EAs was started in the 1990s with analysis how very simple EAs such as the  $(1 + 1)$  EA optimize very simple benchmark problems such as ONEMAX or LEADINGONES [Mühlenbein, 1992; Bäck, 1993; Rudolph, 1997; Droste *et al.*, 2002]. The first runtime analyses of MOEAs followed a similar approach, estimating the runtime of multi-objective analogues of the  $(1 + 1)$  EA such as the SEMO or GSEMO on bi-objective analogues of ONEMAX and LEADINGONES [Laumanns *et al.*, 2002; Giel, 2003; Thierens, 2003]. The analyses of single-objective EAs quickly progressed towards more complex problems such as shortest paths, maximum matchings, the partition problem, the MST problem, and many others [Neumann and Witt, 2010], or more complex algorithms such as the  $(1 + \lambda)$  EA [Jansen *et al.*, 2005],  $(\mu + \lambda)$  EA [Witt, 2006],  $(\mu + \lambda)$  EA [Antipov and Doerr, 2021], and non-elitist algorithms [Dang *et al.*, 2021].

In contrast, due to the more difficult population dynamics of MOEAs, the progress was slower in multi-objective evolutionary computation. There are only few mathematical results on simple MOEAs solving combinatorial optimization problems (to the best of our knowledge only for the MST [Neumann and Wegener, 2007; Neumann and Witt, 2022], shortest path prob-

lems [Horoba, 2009; Neumann and Theile, 2010], and the travelling salesman problem [Lai and Zhou, 2020]). Also, only few results analyze more complex algorithms such as the  $(\mu + 1)$  SIBEA [Brockhoff *et al.*, 2008; Nguyen *et al.*, 2015; Doerr *et al.*, 2016], the MOEA/D [Li *et al.*, 2016; Huang *et al.*, 2019; Huang and Zhou, 2020], the NSGA-II [Zheng *et al.*, 2022; Zheng and Doerr, 2022a; Zheng and Doerr, 2022b; Bian and Qian, 2022; Doerr and Qu, 2023a; Doerr and Qu, 2023b; Doerr and Qu, 2023c; Dang *et al.*, 2023], the NSGA-III [Doerr and Wietheger, 2023], and the SMS-EMOA [Bian *et al.*, 2023]. Very roughly speaking, the works on the NSGA-II show that this algorithm can find the Pareto front of simple bi-objective benchmark problems when the population size is chosen large enough (typically by a constant factor larger than the Pareto front). When the population size is only equal to the size of the Pareto front or when the number of objective is more than two, already on the simple ONEMINMAX benchmark the NSGA-II needs exponential time to find the full Pareto front. So far, no mathematical runtime analysis of the NSGA-II on a combinatorial optimization problem exists.

### 3 Preliminaries: Basic Notation

The Multi-Objective Minimum Spanning Tree problem is stated as follows. Given an input connected graph  $G = (V, E)$ , and a weight function  $w: E \rightarrow \mathbb{N}^d$  on the edges of  $G$ , define the weight of any subgraph  $H$  of  $G$  denoted  $w(H) \in \mathbb{N}^d$  as the sum of the weights of all edges present in  $H$ . We want to find all possible "optimum" subtree weight values  $w \in \mathbb{N}^d$ , in the sense that no subtree of  $G$  has a weight value for which all coordinates are smaller than those of  $w$ , and there exists a spanning tree that has weight  $w$ . Here, we focus on the case where  $d = 2$ . The search space is the set of all subgraphs and is represented with  $S = \{0, 1\}^m$  as a subgraph is a choice of edges. Let  $w = (w_1, w_2): E \rightarrow \mathbb{N}^2$  be the weight function. For a search point  $s$ , we refer to its weight as the geometric point of  $\mathbb{N}^2$  denoted by  $p_s = w(s)$ . We further define

- $w_i^{\max} = \max\{w_i(e), e \in E\}$ , for  $i \in \{1, 2\}$ ,
- $w_{\max} = \max_{i \in \{1, 2\}} w_i^{\max}$
- $w_{\min} = \min_{i \in \{1, 2\}} w_i^{\max}$
- $w_{\text{ub}} = n^2 w_{\max}$ .

As in [Neumann, 2007], the fitness of an individual  $s \in S$  is given by a vector  $f(s) = (f_1(s), f_2(s))$  with

$$f_i(s) = (c(s) - 1)w_{\text{ub}}^2 + (e(s) - (n - 1))w_{\text{ub}} + \sum_{j|s_j=1} w_i(j)$$

for  $i \in \{1, 2\}$ , and where  $w_i(j)$  is the weight of edge  $e_j$  with respect to the function  $w_i$ .  $c(s)$  is the number of connected components in the graph described by  $s$ , and  $e(s)$  is the number of edges in this same graph. Note that for a spanning tree  $s$ ,  $f_i(s) = \sum_{j|s_j=1} w_i(j)$ .

**Definition 1** (Domination). *For  $s, s' \in S$ , we say that  $s$  dominates  $s'$  and we note  $s \preceq s'$  if  $s$  Pareto-dominates*

*$s'$  according to the fitness functions  $f_1$  and  $f_2$ , i.e. if  $f_1(s) \leq f_1(s')$  and  $f_2(s) \leq f_2(s')$ . We say that  $s$  strictly dominates  $s'$  and we note  $s \prec s'$  if  $s \preceq s'$  and  $s$  and  $s'$  have different fitness values.*

Since the relation of domination only depends on the objective value, we also use "domination" to compare objective values rather than individuals.

**Definition 2** (Pareto optimality).  *$s \in S$  is called Pareto optimal if there is no search point  $s' \in S$  that strictly dominates  $s$ . The set of all Pareto optimal search points  $S_{\text{pareto}}$  is called the Pareto set.  $F = f(S_{\text{pareto}})$  is the set of all Pareto optimal objective vectors and is called the Pareto front.*

The goal is to find for each  $q \in F$  of the considered objective function  $f$  an object  $s \in S_{\text{pareto}}$  with  $f(s) = q$ . From now on, as in [Neumann, 2007], we denote by  $\text{conv}(F)$  the lower-left part of the convex hull of  $F$ . The reader may find an illustration of  $\text{conv}(F)$  in appendix of the Arvix version of the paper.

**Definition 3** (Extremal points). *The extremal points of the Pareto front  $F$  are the vertices of the polygonal line forming  $\text{conv}(F)$ .*

Note that for each spanning tree  $T$  on the convex hull there is a  $\lambda \in [0, 1]$  such that  $T$  is a minimum spanning tree with respect to the single weight function  $\lambda w_1 + (1 - \lambda)w_2$  (see e.g. [Knowles and Corne, 2001], [Neumann, 2007]).

Let  $q_1, q_2, \dots, q_r$  be the extremal points sorted in increasing  $f_1$  value.

Observe that  $q_1$  (resp.  $q_r$ ) realises by construction the minimum of  $w_1$  (resp.  $w_2$ ) in  $w(S)$ .

Those points are interesting because they give a solution which is a 2-approximation of the Pareto front ([Neumann, 2007]).

### 3.1 Algorithms

Here, we describe the NSGA-II algorithm. Let  $N$  be the population size and  $n$  the chromosome size. We also give pseudocode for the global SEMO (GSEMO), since part of our results directly come from the study of this algorithm in [Neumann and Wegener, 2007], and since most of these results also apply to this algorithm.

#### The NSGA-II

The NSGA-II is an evolutionary algorithm, which means it maintains a population of  $N \in \mathbb{N}$  solutions to an optimization problem, named *individuals* (here, subgraphs of  $G$ ). This population evolves at each over multiple generations. In each iteration, the algorithm generates an offspring population from selected parents in the current population, using some reproduction and mutation mechanisms. Note however that these mechanisms are not intrinsic to the NSGA-II and have to be specified for the task at hand. From the combined parent and offspring population, the algorithm decides which individuals to keep according to their fitness. The NSGA-II is centered around two notions. First, *rank*, which defines

how good an individual is in the current population, and second, *crowding distance*, which quantifies how much diversity it brings to the population.

**Definition 4 (Rank).** *Let  $X$  be a population. We recursively define the rank of every individual in  $X$ . An individual that is not strictly dominated by any individual in  $X$  has rank 1. Then, if ranks  $1, \dots, k$  are defined, an individual that is not strictly dominated by any individual of  $X$ , except those of rank  $1, \dots, k$ , and that is dominated by at least one individual of rank  $k$ , has rank  $k + 1$ .*

The ranks of individuals in a given population  $X$  can be computed in quadratic time using the fast-non-dominated-sort algorithm. The reader will find in appendix of the Arvix version of the paper the associated pseudocode.

We now introduce the notion of crowding distance.

**Definition 5 (Crowding distance).** *Let  $X$  be a finite set, and  $f: X \rightarrow \mathbb{R}$ . Let  $x_1, \dots, x_N$  be the elements of  $X$ , sorted in increasing order of  $f$  values. Then, the crowding distance of each  $x_i$  is defined as*

$$cDis(x_i) = \begin{cases} +\infty & \text{if } i = 1 \text{ or } i = n \\ \frac{f(x_{i+1}) - f(x_i)}{f(x_i) - f(x_1)} & \text{otherwise.} \end{cases} \quad (1)$$

Note that in our case, we are dealing with multiple fitness functions  $(f_1, f_2)$ . The crowding distance of an element is then the sum of the crowding distances of this element for each function. Computing the crowding distance of each individual in a population of size  $N$  can be done naively with two sortings and  $O(N)$  subtractions and divisions. It can then be done in  $O(N \log(N))$ , which is negligible compared to fast-non-dominated sort.

The pseudocode for NSGA-II can be found in Algorithm 1.

---

**Algorithm 1:** The NSGA-II.

---

```

1 Generate initial population  $P \in (\{0, 1\}^m)^N$ 
2 repeat
3   Generate offspring population  $Q \in (\{0, 1\}^m)^N$ 
4   Let  $R = P \cup Q$ 
5   Sort  $R$  with fast-non-dominated-sort to get
   the sets  $F_i, i \in \mathbb{N}$  of the individuals of rank  $i$ 
6   Find  $i_{cut} = \max\{i \mid \sum_{k=0}^{i-1} |F_k| < N\}$ 
7   Calculate crowding distance of each individual
   in  $F_{i_{cut}}$ 
8   Let  $\widehat{F}_{i_{cut}}$  be the  $N - \sum_{k=0}^{i_{cut}-1} |F_k|$  individuals
   in  $F_{i_{cut}}$  with largest crowding distance,
   chosen at random in case of a tie
9    $P = (\bigcup_{i=0}^{i_{cut}-1} F_i) \cup \widehat{F}_{i_{cut}}$ 
10 until forever;

```

---

**The Global Simple Evolutionary Multiobjective Optimizer (GSEMO)**

GSEMO is also an evolutionary algorithm, studied on the mc-MST problem in [Neumann, 2007] and [Neumann

and Wegener, 2007]. Its functioning is simpler than that of the NSGA-II. It generates one individual  $s$  at each generation, adding it to the population if it is not dominated by any other individual, and removes those that are dominated by  $s$ . A pseudocode for GSEMO, as described in [Neumann, 2007] can be found in Algorithm 2.

---

**Algorithm 2:** GSEMO

---

```

1 Generate initial population  $P$ , which consists of a
  unique individual  $s \in \{0, 1\}^n$ , chosen randomly.
2 repeat
3   Choose a random  $s \in P$ 
4   Generate an offspring  $s'$  from  $x$ , flipping each
   bit with probability  $\frac{1}{n}$ .
5   if no individual in  $P$  dominates  $s'$  then
6     Add  $s'$  to  $P$ 
7     Remove all individuals in  $P$  that  $s'$ 
   dominates
8 until forever;

```

---

## 4 Analysis of the GSEMO and the NSGA-II on the Bi-Objective Minimum Spanning Tree Problem

In this section, we prove two main results on the expected runtime of the GSEMO and the NSGA-II on the Bi-Objective Minimum Spanning Tree Problem.

To state our theorem on the NSGA-II in the strongest possible form, we need to introduce two parameters which depend on the offspring generation mechanism which the NSGA-II works with.

For any population  $P$  containing no spanning tree, any  $s \in F_1$  and any position  $i \in \{1, \dots, n\}$  of a bit of value 0, let  $\mathbf{p}_1$  be a lower bound (that does not depend on  $P, s, i$  and  $j$ ) of the probability that there exists in the offspring a child generated with  $s$  as (one of) the parent(s), that differs from  $s$  on exactly bit  $i$ .

For any population  $P$  containing at least a spanning tree, any  $s \in F_1$ , and any pair of bits of different value  $i, j \in \{1, \dots, n\}$ , let  $\mathbf{p}_2$  be a lower bound (that does not depend on  $P, s, i$  and  $j$ ) of the probability that there exists in the offspring a child generated with  $s$  as (one of) the parent(s), that differs from  $s$  on exactly bits  $i, j$ . In Section 5, we will detail why such precise definitions might reveal useful.

**Theorem 6.** *The expected number of generations until the NSGA-II, working on the fitness function  $f$  with a population of size  $N \geq 4((n-1)w_{\min} + 1)$  and with any offspring generation mechanism resulting in  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , constructs a population which includes a spanning tree for each extremal vector of  $\text{conv}(F)$  is upper bounded by  $O(\frac{\log n}{\mathbf{p}_1} + \frac{\log(nw_{\max})}{\mathbf{p}_2})$ .*

**Theorem 7.** *The expected number of fitness evaluations until the GSEMO working on the fitness function  $f$  constructs a population, which includes a spanning tree*

for each extremal vector of  $\text{conv}(F)$  is upper bounded by  $O(m^2 n w_{\min} \log(n w_{\max}))$ .

The proof is split into three parts. First, we prove that the NSGA-II and the GSEMO find spanning trees in negligible time. Secondly, we introduce an elitism property of the NSGA-II from the moment when the population includes spanning trees. Then, we bound the time taken by both NSGA-II and GSEMO to find the extremal points from this point.

#### 4.1 Sampling the First Spanning Tree

Regarding the first phase of the optimization process, the proof of [Neumann, 2007, Lemma 5] bounds the expected time until the population of the GSEMO contains at least one spanning tree from above, giving the following lemma.

**Lemma 8.** *The GSEMO working on the fitness function  $f$  constructs a population with at least one spanning tree in expected time  $O(m \log n)$ .*

We argue that a similar bound holds for the NSGA-II.

**Lemma 9.** *The NSGA-II working on the fitness function  $f$  constructs a population with at least one spanning tree in expected time  $O(\frac{\log n}{p_1})$ .*

The proof is essentially an adaptation of [Neumann, 2007, Lemma 5], the details of which are in the appendix of the Arvix version of this paper. We first bound the time before the population contains a connected graph: to do so, we observe that for a subgraph  $H$  with  $l \geq 2$  connected components, there are at least  $l - 1$  edges of  $G$  that decrease  $l$ . This allows us to use  $p_1$  to give a lower bound for the probability that these specific edges are added, and thus that connected components merge fast enough. A similar argument is used for the second phase, to show that excessive edges are deleted at roughly the same speed.

#### 4.2 An Elitism Property of the NSGA-II

One of the observations that make the study of the GSEMO easier is that it has an elitism property: an individual will not disappear from the population unless it is replaced by a dominating one. Here, we introduce a lemma, inspired by [Zheng and Doerr, 2022a], which shows that under a certain condition on the population size, the NSGA-II has a similar property.

**Lemma 10.** *Let  $P$  be a population such that  $|P| > 4((n - 1)w_{\min} + 1)$  and having at least one spanning tree. Let  $P'$  be the next population. For each individual  $s$  in  $P$ , there is an individual  $s'$  in  $P'$  such that  $s' \preceq s$ .*

The proposed proof for this lemma requires to introduce the notion of incomparable set, which is used to bound the size of  $F_1$ .

**Definition 11** (Incomparable set).  *$A \subseteq S$  is an incomparable set if there is no pair of individuals  $s, s' \in A$ , such that  $s' \prec s$ .*

We now make the following observation regarding the size of incomparable sets.

**Lemma 12.** *Let  $P$  be an incomparable set containing at least one spanning tree. Then the number of objective values of  $P$  is bounded by  $(n - 1)w_{\min} + 1$ .*

*Proof.* Since  $f$  is designed such that a spanning tree strictly dominates every non-spanningtree graph,  $P$  consists of spanning trees only. Among two spanning trees of  $P$  with the same  $f_1$  value, one of them necessarily dominates the other.  $P$  being an incomparable set, this does only happens if the two spanning trees have the same objective value. Thus, for a fixed value of  $f_1$ , there is at most one point in  $f(P)$  having this  $f_1$  value. Since a spanning tree has  $n - 1$  edges, and since by definition  $0 \leq w_1(e) \leq w_1^{\max}$  for any edge  $e$ , we deduce that, for any spanning tree  $s$ ,  $0 \leq f_1(s) \leq (n - 1)w_1^{\max}$ . It follows that

$$|w(P)| \leq |f_1(ST)| \leq (n - 1)w_1^{\max} + 1,$$

where  $ST$  is the set of elements of  $S$  representing a spanning tree constructed from  $G$ . We do the same with  $f_2$ , which gives  $|w(P)| \leq (n - 1)w_{\min} + 1$ , this bound being the minimum of the two bounds previously obtained.  $\square$

The subsequent formal proof for Lemma 10 is rather long, and may be found in appendix of the Arvix version of the paper. However, we give a sketch of the principal arguments: we observe that, by definition of the fronts, the lemma requires to be proven only for individuals of the first front. Then, we show that, for a given fitness value  $p$ , there are at most 4 individuals that have fitness  $p$  and positive crowding distance. Finally, we use Lemma 12 to conclude that, because the NSGA-II will first choose individuals with positive crowding distance, the size of the population is such that the new generation will represent all fitness values of the first front.

#### 4.3 Sampling the Extremal Points of the Pareto Front

This section gives upper bounds for the expected run-times of the two algorithms, assuming the population contains at least a spanning tree and therefore proves Theorem 6 and 7.

We build our analysis on the fact that the extremal points are the unique minimums for specific linear combinations of the weights  $w_1$  and  $w_2$ . The following lemma gives the combination for  $q_i$ , when  $1 \leq i \leq r$ .

**Lemma 13.** *Let  $i \in \{1, \dots, r\}$  and let for all objective values  $p$ ,*

$$d_i(p) := (w_1(q_{i-1}) - w_1(q_{i+1}))(w_2(p) - w_2(q_i)) \\ + (w_2(q_{i+1}) - w_2(q_{i-1}))(w_1(p) - w_1(q_i)).$$

*Then, for a given objective value  $p = w(s)$  of some individual  $s$ ,  $d_i(p) \geq 0$  and  $d_i(p) = 0 \Leftrightarrow p = q_i$ .*

*Proof.* As shown in Figure 1, the set of all objective values is contained in the intersection of the upper-half planes  $H_j$  of the lines  $\overline{q_j q_{j+1}}$ ,  $1 \leq j \leq r - 1$ . For any such  $j$ , let

$$v_j = \left( \frac{w_2(q_j) - w_2(q_{j+1})}{2} \right. \\ \left. \frac{w_1(q_{j+1}) - w_1(q_j)}{2} \right).$$

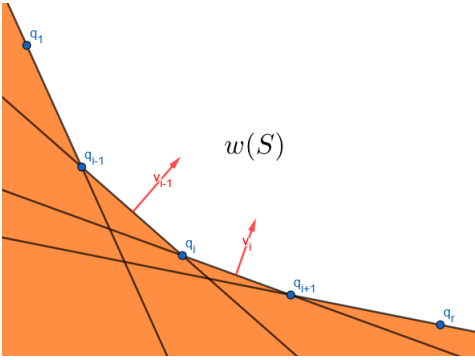


Figure 1: Position of  $w(S)$  relative to the  $\overline{q_i q_{i+1}}$

This vector is normal to the line  $\overline{q_j q_{j+1}}$  and is pointing towards  $w(S)$ . Thus, for all  $p \in \mathbb{R}^2, p \in H_j$  if and only if  $p \cdot v_j \geq q \cdot v_j$  for any point  $q$  lying on  $\overline{q_j q_{j+1}}$ , with equality if and only if  $p \in (q_j, q_{j+1})$ . Now let us fix  $i \in \{2, \dots, r-1\}$ . We know that  $q_i$  lies on both  $(q_i, q_{i+1})$  and  $(q_{i-1}, q_i)$ , so, for all  $p \in w(S), p \cdot v_i \geq q_i \cdot v_i$  and  $p \cdot v_{i-1} \geq q_i \cdot v_{i-1}$ , with equality if and only if  $p \in (q_i, q_{i+1})$  or  $p \in (q_{i-1}, q_i)$ , respectively. By summing these two inequalities, we get  $(p - q_i) \cdot (v_i + v_{i-1}) \geq 0$ , that is  $d_i(p) \geq 0$ , with equality if and only if  $p \in (q_i, q_{i+1}) \cap (q_i, q_{i-1})$ . Since  $q_i$  and  $q_{i+1}$  are extremal points, these two lines cannot be parallel, hence  $(q_i, q_{i+1}) \cap (q_i, q_{i-1}) = \{q_i\}$ , which concludes the proof.  $\square$

We suppose from now on that the population  $P$  contains a spanning tree, which is true after an expected number of  $O(m \log n)$  iterations by Lemmas 8 and 9.

For all  $i \in \{2, \dots, r-1\}$ , we define  $d_i(p)$  as stated in Lemma 13. The lemma further suggests that we define

$$\begin{aligned} d_1(p) &= w_1(p) - w_1(q_1), \\ d_r(p) &= w_2(p) - w_2(q_r). \end{aligned}$$

To prove the claimed upper bound, we use multiplicative drift analysis [Doerr *et al.*, 2012] on a quantity derived from these functions. A reminder for this technique may be found in appendix of the Arvix version of the paper.

To use the multiplicative drift theorem, we introduce a potential that has a multiplicative drift. For a given population  $P$ , let  $d_i(P) = \min_{p \in w(P)} d_i(p)$  and  $d(P) = \sum_{i=1}^r d_i(P)$ .

**Lemma 14.** *Let  $P(t)$  be the population of the NSGA-II or the GSEMO algorithm after  $t$  iterations. Then  $d_i(P(t))$  for any  $1 \leq i \leq r$ , and  $d(P(t))$  are nonnegative integers and are non-increasing with respect to  $t$ .*

*Proof.* Let  $p_i(t) = \arg \min_{p \in w(P(t))} d_i(p)$ . Using Lemma 10 in the case of the NSGA-II, and by the definition in the case of the GSEMO, there exists an element  $p'_i \in w(P(t+1))$  such that  $p'_i \leq p_i$ . Hence,

$$\min_{p' \in w(P(t+1))} d_i(p') \leq d_i(p'_i) \leq d_i(p_i) = \min_{p \in w(P(t))} d_i(p),$$

so  $d_i(P(t))$  is non-increasing. Lemma 13 gives that  $d_i(P(t))$  is nonnegative. Summing  $d_i$  for  $1 \leq i \leq r$  gives the same two properties for  $d(P(t))$ .  $\square$

In order to apply drift analysis, we need another lemma given by [Neumann and Wegener, 2007].

**Lemma 15** ([Neumann and Wegener, 2007], Lemma 2). *Let  $\tilde{w}: S \rightarrow \mathbb{R}$  be any function which is a linear combination with nonnegative coefficients of  $w_1$  and  $w_2$ . Let  $s$  be a search point describing a spanning tree  $T$ . Let  $\tilde{w}_{opt}$  be the minimum value taken by  $\tilde{w}$  on spanning trees. Then there exists a set of  $n$  2-bit flips resulting on new spanning trees, such that the average weight decrease of these flips is at least  $\frac{\tilde{w}(s) - \tilde{w}_{opt}}{n}$ .*

We now prove the following lemma:

**Lemma 16.** *The expected number of generations until the NSGA-II (resp. GSEMO), working on the fitness function  $f$  with a population of size  $N \geq 4((n-1)w_{\min} + 1)$  and with any offspring generation mechanism (resp. its intrinsic mechanism), constructs a population  $P$  such that  $d(P) = 0$  is upper bounded by  $O(\frac{\log(nw_{\max})}{\mathbf{p}_2})$  (resp.  $O(m^2 w_{\min} \log(nw_{\max}))$ ).*

*Proof.* For any  $i \in \{1, \dots, r\}$  and any iteration  $t$ , let  $s_{i,t}$  be an individual such that  $d_i(s_{i,t}) = d_i(P(t))$ . Then, let  $\mathbf{p}_{2,(i,t)}$  be a lower bound over all pairs of bit positions of the probability that there exists, in the offspring, a child generated with  $s_{(i,t)}$  as (one of) the parent(s) that differs from  $s_{(i,t)}$  on exactly this pair of bits. For the NSGA-II algorithm, by definition of  $\mathbf{p}_2$ , we have  $\mathbf{p}_{2,(i,t)} \geq \mathbf{p}_2$ .

For the GSEMO algorithm, if there is at least one spanning tree in the population, the size of the population is upper-bounded by  $(n-1)w_{\min} + 1$  by Lemma 12. Therefore,

$$\mathbf{p}_{2,(i,t)} \geq \frac{1}{(n-1)w_{\min} + 1} \cdot \frac{(1 - \frac{1}{m})^{m-2}}{m^2} \geq \frac{1}{((n-1)w_{\min} + 1)em^2}$$

We also denote  $\mathbf{p}_2 = \frac{1}{(n-1)w_{\min}em^2}$  when analysing GSEMO, such that we have  $\mathbf{p}_{2,(i,t)} \geq \mathbf{p}_2$  for both algorithms.

We now apply drift analysis on  $d(P(t))$ . Indeed, for  $i \in \{1, \dots, r\}$ ,  $d_i(p)$  is a linear combination of  $w_1$  and  $w_2$  with non-negative coefficients minimum  $d_i^{opt} = 0$ .

Focusing only on the set of  $n$  2-bits flips given by Lemma 15 on a search point being one of  $\arg \min_{s \in P} d_i(s)$ , and noticing that, since  $d_i$  is non-increasing (Lemma 14), all other bit flips contribute positively to the drift, we have:

$$\mathbb{E}[d_i(P(t)) - d_i(P(t+1)) | d_i(P(t)) = x] \geq (x/n) \cdot n \cdot \mathbf{p}_{2,(i,t)}.$$

Using this estimate, we show the following lower bound for the drift of our potential:

$$\mathbb{E}[d(P(t)) - d(P(t+1)) | d(P(t)) = x] \geq x \cdot \mathbf{p}_2.$$

Let  $T = \min\{t \in \mathbb{N} \mid d(P(t)) = 0\}$ . Since  $d(P)$  is an non-negative integer, the minimum strictly positive value of  $d(P)$  is 1. Since each edge has a weight at

most  $2w_{\max}^2$ ,  $d(P) \leq \sum_{i=1}^r 2mw_{\max}^2 \leq 2rn^2w_{\max}^2$ . Finally, since the convex hull is an incomparable set, we have  $r \leq (n-1)w_{\min} + 1$  by Lemma 12. Hence we derive the following upper bound for  $d(P(t))$ :

$$\max_t d(P(t)) \leq ((n-1)w_{\min} + 1)mw_{\max} = O(n^3w_{\max}w_{\min}).$$

Using multiplicative drift theorem, we obtain

$$\mathbb{E}(T) \leq \frac{1}{\mathbf{p}_2} \log \frac{\max_{t \in \mathbb{N}} d(P(t))}{1} = O\left(\frac{\log n + \log w_{\max}}{\mathbf{p}_2}\right).$$

We then conclude the proof by plugging in the value of  $\mathbf{p}_2$  for GSEMO, which does only one fitness evaluation per iteration.  $\square$

Combining the last lemma with Lemma 13, and noticing that NSGA-II and GSEMO do respectively  $N$  and 1 fitness evaluation(s) per generation, we conclude the proof of theorems 6 and 7.

## 5 Study of Offspring Generation Mechanisms of the NSGA-II

This section is dedicated to demonstrating applications of Theorem 6 on the study of the influence of particular offspring generation mechanisms on the expected runtime of the NSGA-II before finding the extremal points.

We start with a very rudimentary mechanism, composed of fair selection (selecting every individual  $s$  in  $P$  as a parent), and standard bitwise mutation with constant  $c$ ,  $0 \leq c \leq m$  (mutating every bit of a given parent with probability  $\frac{c}{m}$ ). Then, we can set  $\mathbf{p}_1 = \frac{c}{m}(1 - \frac{c}{m})^{m-1} \geq \frac{c}{ecm}$ , and  $\mathbf{p}_2 = \frac{c^2}{m^2}(1 - \frac{c}{m})^{m-2} \geq \frac{c^2}{ecm^2}$ . Theorem 6 then gives the following corollary.

**Corollary 17.** *The expected time until the NSGA-II, working on the fitness function  $s$  with a population of size  $N \geq 4((n-1)w_{\min} + 1)$  and using standard bitwise mutation and fair selection, constructs a population which includes a spanning tree for each extremal vector of  $\text{conv}(F)$  is upper bounded by  $O(m^2 \log(nw_{\max}))$  generations, and  $O(m^2 N \log(nw_{\max}))$  fitness evaluations.*

Theorem 6 also ensures that adding crossover with constant probability  $0 < q < 1$  does not worsen the asymptotic complexity of the NSGA-II. To be more precise, let us consider a random monoparental reproduction scheme  $\mathcal{M}$ , which, from an individual  $s \in S$ , generates an offspring  $\mathcal{M}(s)$ . From this operator, and a given crossover operator  $\mathcal{C}$  we derive a biparental reproduction scheme  $\mathcal{M}'$ , which, from a pair of individuals  $s, s'$ , generates, with probability  $q$ , the children  $\mathcal{C}(s, s')$  and  $\mathcal{C}(s', s)$ , and with probability  $(1-q)$ , the children  $\mathcal{M}(s)$ , and  $\mathcal{M}(s')$ . Now, consider an offspring generation scheme which puts the selected parents into pairs and applies  $\mathcal{M}'$  to these pairs. Finally, note that if  $\mathbf{p}_1, \mathbf{p}_2$  and  $\mathbf{p}'_1, \mathbf{p}'_2$  are the pairs of probabilities defined in 6 for the offspring generation mechanism using  $\mathcal{M}$  or  $\mathcal{M}'$ , then we have trivially  $\mathbf{p}'_1 \geq (1-q)\mathbf{p}_1$  and  $\mathbf{p}'_2 \geq (1-q)\mathbf{p}_2$ . This gives the following corollary.

**Corollary 18.** *For a given NSGA-II implementation, replacing  $\mathcal{M}$  by  $\mathcal{M}'$ , as described above (i.e. adding crossover with constant probability) does not worsen the asymptotic expected runtime (up to a multiplicative constant).*

Finally, to demonstrate the relevance of the  $\mathbf{p}_2$  parameter, let us introduce a problem-specific “balanced” mutation operator, described in [Neumann and Wegener, 2007]. For any tree individual  $s \in S$  the mutation operator flips each 1-bit with probability  $\frac{1}{n-1}$ , and each 0-bit with probability  $\frac{1}{m-n+1}$ , and applies standard bitwise mutation otherwise. We use fair selection for this example. By definition, the value of  $\mathbf{p}_1$  remains unchanged compared to standard bitwise mutation, that is,  $O(\frac{1}{m})$ . However, one can show that  $\mathbf{p}_2$  is increased from  $O(\frac{1}{m^2})$  to  $O(\frac{1}{(m-n)n})$  (see appendix of the Arvix version of the paper). Theorem 6 automatically gives:

**Corollary 19.** *The expected runtime of the NSGA-II, working on fitness function  $f$ , with  $|P| > 4((n-1)w_{\min} + 1)$ , using fair selection and balanced mutation is  $O((m-n)n \log(nw_{\max}) + m \log n)$  generations and  $O(N((m-n)n \log(nw_{\max}) + m \log n))$  fitness evaluations.*

## 6 Conclusion

In this first mathematical runtime analysis of the NSGA-II on a combinatorial optimization problem, we provided a general approach to proving runtime guarantees for MOEAs solving the bi-objective MST problem. For the global SEMO, this gave a bounds lower than the previously known ones by a factor of  $\Omega(|F|)$ . More interestingly, we could prove the same performance guarantees for the much more complex NSGA-II. Our result applies to several variants of the NSGA-II, including some that use crossover. As for the simple global SEMO algorithm, we obtain better guarantees when employing a balanced mutation operator (which supports the general belief that analyses on simple toy algorithms can nevertheless give useful hint for the use of more complex algorithms).

Overall, this work indicate that mathematical runtime analyses for the NSGA-II are possible also for combinatorial optimization problems. In this first work, we mostly concentrated on proving performance guarantees at all. For future work, it would be interesting to derive more insights on how to optimally use the NSGA-II on the particular problem (we only saw that balanced mutation is preferable). That such results are possible in principle is again indicated by the simpler works on artificial benchmarks, where, e.g., [Doerr and Qu, 2023a] gave some indications on the right mutation rate. Clearly, runtime analyses of the NSGA-II on other combinatorial optimization problems would also be desirable to put this research direction on a broader basis.

## Acknowledgments

This work was supported by a public grant as part of the Investissements d’avenir project, reference ANR-11-



LABX-0056-LMH, LabEx LMH and a fellowship via the International Exchange Program of École Polytechnique.

## References

- [Antipov and Doerr, 2021] Denis Antipov and Benjamin Doerr. A tight runtime analysis for the  $(\mu + \lambda)$  EA. *Algorithmica*, 83:1054–1095, 2021.
- [Arroyo *et al.*, 2008] José Elias Claudio Arroyo, Pedro Sampaio Vieira, and Dalessandro Soares Vianna. A GRASP algorithm for the multi-criteria minimum spanning tree problem. *Annals of Operations Research*, 159:125–133, 2008.
- [Auger and Doerr, 2011] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics*. World Scientific Publishing, 2011.
- [Bäck, 1993] Thomas Bäck. Optimal mutation rates in genetic search. In *International Conference on Genetic Algorithms, ICGA 1993*, pages 2–8. Morgan Kaufmann, 1993.
- [Bian and Qian, 2022] Chao Bian and Chao Qian. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *Parallel Problem Solving From Nature, PPSN 2022*, pages 428–441. Springer, 2022.
- [Bian *et al.*, 2023] Chao Bian, Yawen Zhou, Miqing Li, and Chao Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, 2023. To appear.
- [Bossek and Grimme, 2017] Jakob Bossek and Christian Grimme. A Pareto-beneficial sub-tree mutation for the multi-criteria minimum spanning tree problem. In *Symposium Series on Computational Intelligence, SSCI 2017*, pages 1–8. IEEE, 2017.
- [Brockhoff *et al.*, 2008] Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature, PPSN 2008*, pages 651–660. Springer, 2008.
- [Dang *et al.*, 2021] Duc-Cuong Dang, Anton V. Ermeev, and Per Kristian Lehre. Escaping local optima with non-elitist evolutionary algorithms. In *AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 12275–12283. AAAI Press, 2021.
- [Dang *et al.*, 2023] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 2023. To appear.
- [Deb *et al.*, 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
- [Doerr and Qu, 2023a] Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *Transactions on Evolutionary Computation*, 2023. <https://doi.org/10.1109/TEVC.2023.3250552>.
- [Doerr and Qu, 2023b] Benjamin Doerr and Zhongdi Qu. From understanding the population dynamics of the NSGA-II to the first proven lower bounds. In *Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 2023. To appear.
- [Doerr and Qu, 2023c] Benjamin Doerr and Zhongdi Qu. Runtime analysis for the NSGA-II: Provable speed-ups from crossover. In *Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 2023. To appear.
- [Doerr and Wietheger, 2023] Benjamin Doerr and Simon Wietheger. A mathematical runtime analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In *International Joint Conference on Artificial Intelligence, IJCAI 2023*. ijcai.org, 2023. To appear.
- [Doerr *et al.*, 2012] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64:673–697, 2012.
- [Doerr *et al.*, 2016] Benjamin Doerr, Wanru Gao, and Frank Neumann. Runtime analysis of evolutionary diversity maximization for OneMinMax. In *Genetic and Evolutionary Computation Conference, GECCO 2016*, pages 557–564. ACM, 2016.
- [Droste *et al.*, 2002] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.
- [Ehrgott, 2005] Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.
- [Giel, 2003] Oliver Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Congress on Evolutionary Computation, CEC 2003*, pages 1918–1925. IEEE, 2003.
- [Hamacher and Ruhe, 1994] Horst W. Hamacher and Günther Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.
- [Horoba, 2009] Christian Horoba. Analysis of a simple evolutionary algorithm for the multiobjective shortest path problem. In *Foundations of Genetic Algorithms, FOGA 2009*, pages 113–120. ACM, 2009.
- [Huang and Zhou, 2020] Zhengxin Huang and Yuren Zhou. Runtime analysis of somatic contiguous hypermutation operators in MOEA/D framework. In *Conference on Artificial Intelligence, AAAI 2020*, pages 2359–2366. AAAI Press, 2020.
- [Huang *et al.*, 2019] Zhengxin Huang, Yuren Zhou, Zefeng Chen, and Xiaoyu He. Running time analysis of MOEA/D with crossover on discrete optimization

- problem. In *Conference on Artificial Intelligence, AAAI 2019*, pages 2296–2303. AAAI Press, 2019.
- [Jansen *et al.*, 2005] Thomas Jansen, Kenneth A. De Jong, and Ingo Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13:413–440, 2005.
- [Knowles and Corne, 2000] Joshua D. Knowles and David Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8:149–172, 2000.
- [Knowles and Corne, 2001] Joshua D. Knowles and David W. Corne. A comparison of encodings and algorithms for multiobjective minimum spanning tree problems. In *Congress on Evolutionary Computation, CEC 2001*, pages 544–551. IEEE, 2001.
- [Lai and Zhou, 2020] Xinsheng Lai and Yuren Zhou. Analysis of multiobjective evolutionary algorithms on the biobjective traveling salesman problem (1, 2). *Multimedia Tools and Applications*, 79:30839–30860, 2020.
- [Laumanns *et al.*, 2002] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10:263–282, 2002.
- [Li *et al.*, 2016] Yuan-Long Li, Yu-Ren Zhou, Zhi-Hui Zhan, and Jun Zhang. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20:563–576, 2016.
- [Majumder *et al.*, 2020] Saibal Majumder, Mohuya B. Kar, Samarjit Kar, and Tandra Pal. Uncertain programming models for multi-objective shortest path problem with uncertain parameters. *Soft Computing*, 24:8975–8996, 2020.
- [Mühlenbein, 1992] Heinz Mühlenbein. How genetic algorithms really work: mutation and hillclimbing. In *Parallel Problem Solving from Nature, PPSN 1992*, pages 15–26. Elsevier, 1992.
- [Neumann and Theile, 2010] Frank Neumann and Madeleine Theile. How crossover speeds up evolutionary algorithms for the multi-criteria all-pairs-shortest-path problem. In *Parallel Problem Solving from Nature, PPSN 2010, Part I*, pages 667–676. Springer, 2010.
- [Neumann and Wegener, 2007] Frank Neumann and Ingo Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science*, 378:32–40, 2007.
- [Neumann and Witt, 2010] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.
- [Neumann and Witt, 2022] Frank Neumann and Carsten Witt. Runtime analysis of single- and multi-objective evolutionary algorithms for chance constrained optimization problems with normally distributed random variables. In *International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 4800–4806. ijcai.org, 2022.
- [Neumann, 2007] Frank Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181:1620–1629, 2007.
- [Nguyen *et al.*, 2015] Anh Quang Nguyen, Andrew M. Sutton, and Frank Neumann. Population size matters: rigorous runtime results for maximizing the hypervolume indicator. *Theoretical Computer Science*, 561:24–36, 2015.
- [Parraga-Alava *et al.*, 2017] Jorge Parraga-Alava, Márcio Dorn, and Mario Inostroza-Ponta. Using local search strategies to improve the performance of NSGA-II for the multi-criteria minimum spanning tree problem. In *Congress on Evolutionary Computation, CEC 2017*, pages 1119–1126. IEEE, 2017.
- [Rudolph, 1997] Günter Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kováč, 1997.
- [Thierens, 2003] Dirk Thierens. Convergence time analysis for the multi-objective counting ones problem. In *Evolutionary Multi-Criterion Optimization, EMO 2003*, pages 355–364. Springer, 2003.
- [Witt, 2006] Carsten Witt. Runtime analysis of the  $(\mu + 1)$  EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14:65–86, 2006.
- [Zheng and Doerr, 2022a] Weijie Zheng and Benjamin Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, pages 611–619. ACM, 2022.
- [Zheng and Doerr, 2022b] Weijie Zheng and Benjamin Doerr. Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for three or more objectives. *CoRR*, abs/2211.13084, 2022.
- [Zheng *et al.*, 2022] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Conference on Artificial Intelligence, AAAI 2022*, pages 10408–10416. AAAI Press, 2022.
- [Zhou *et al.*, 2011] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagarathnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1:32–49, 2011.