Runtime Analysis of Single- and Multi-Objective Evolutionary Algorithms for Chance Constrained Optimization Problems with Normally Distributed Random Variables

Frank Neumann[1] , **Carsten Witt**[2]

[1]Optimisation and Logistics, University of Adelaide, Adelaide, Australia
[2]DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

Chance constrained optimization problems allow to model problems where constraints involving stochastic components should only be violated with a small probability. Evolutionary algorithms have been applied to this scenario and shown to achieve high quality results. With this paper, we contribute to the theoretical understanding of evolutionary algorithms for chance constrained optimization. We study the scenario of stochastic components that are independent and Normally distributed. Considering the simple single-objective (1+1) EA, we show that imposing an additional uniform constraint already leads to local optima for very restricted scenarios and an exponential optimization time. We therefore introduce a multi-objective formulation of the problem which trades off the expected cost and its variance. We show that multi-objective evolutionary algorithms are highly effective when using this formulation and obtain a set of solutions that contains an optimal solution for any possible confidence level imposed on the constraint. Furthermore, we prove that this approach can also be used to compute a set of optimal solutions for the chance constrained minimum spanning tree problem. Experimental investigations on instances of the NP-hard stochastic minimum weight dominating set problem confirm the benefit of the multi-objective approach in practice.

1 Introduction

Many real-world optimization problems involve solving optimization problems that contain stochastic components [Ben-Tal *et al.*, 2009]. Chance constraints [Charnes and Cooper, 1959] allow to limit the probability of violating a constraint involving stochastic components. In contrast to limiting themselves to ruling out constraint violations completely, this allows to deal with crucial constraints in a way that allows to ensure meeting the constraints with high confidence (usually determined by a confidence level α) while still maintaining solutions of high quality with respect to the given objective function.

Evolutionary algorithms have successfully been applied chance constrained optimization problems [Poojari and Varghese, 2008; Liu *et al.*, 2013]. Recent studies investigated the classical knapsack problem in static [Xie *et al.*, 2019; Xie *et al.*, 2020; Neumann *et al.*, 2022] and dynamic settings [Assimi *et al.*, 2020] as well as complex stockpile blending problems [Xie *et al.*, 2021a] and the optimization of submodular functions [Neumann and Neumann, 2020]. Theoretical analyses for submodular problems with chance constraints, where each stochastic component is uniformly distributed and has the same amount of uncertainty, have shown that greedy algorithms and evolutionary Pareto optimization approaches only lose a small amount in terms of approximation quality when comparing against the corresponding deterministic problems [Doerr *et al.*, 2020; Neumann and Neumann, 2020] and that evolutionary algorithms significantly outperform the greedy approaches in practice. Other recent theoretical runtime analyses of evolutionary algorithms have produced initial results for restricted classes of instances of the knapsack problem where the weights are chosen randomly [Neumann and Sutton, 2019; Xie *et al.*, 2021b].

For our theoretical investigations, we use runtime analysis which is a major theoretical tool for analyzing evolutionary algorithms in discrete search spaces [Neumann and Witt, 2010; Jansen, 2013; Doerr and Neumann, 2020]. In order to understand the working behaviour of evolutionary algorithms on broader classes of problems with chance constraints, we consider the optimization of linear functions with respect to chance constraints where the stochastic components are independent and each weight w_i is chosen according to a Normal distribution $N(\mu_i, \sigma_i^2)$. This allows to reformulate the problem by a deterministic equivalent non-linear formulation involving a linear combination of the expected expected value and the standard deviation of a given solution.

We investigate how evolutionary algorithms can deal with chance constrained problems where the stochastic elements follow a Normal distribution. We first analyze the classical (1+1) EA on this problem formulation and we show that imposing a simple cardinality constraint for a simplified class of instances leads to local optima and exponential lower bounds for the (1+1) EA. In order to deal with the issue of the (1+1) EA not being able to handle even simple constraints due to the non-linearity of the objective functions, we introduce a Pareto optimization approach for the chance constrained optimization problems under investigation. So far, Pareto optimization approaches that achieved provably good solution provided a trade-off with respect to the original objective functions and given constraints. In contrast to this, our approach trades off the different components determining the uncertainty of solutions, namely the expected value and variance of a solution. A crucial property of our reformulation is that the extreme points of the Pareto front provide optimal solutions for any linear combination of the expected value and the standard deviation and solves the original chance constrained problem for any confidence level $\alpha \geq 1/2$. These insights mean that the users of the evolutionary multi-objective algorithm does not need to know the desired confidence level in advance, but can pick from a set of trade-offs with respect to the expected value and variance for all possible confidence levels. We show that this approach can also be applied to the chance constrained minimum spanning tree problem where each edge cost is chosen independently according to its own Normal distribution. In terms of algorithms, we analyze the well-known GSEMO [Giel, 2003] which has been frequently applied in the context of Pareto optimization [Neumann and Wegener, 2006; Kratsch and Neumann, 2013; Friedrich and Neumann, 2015; Zhou *et al.*, 2019] and show that it computes such an optimal set of solutions for any confidence level of $\alpha \geq 1/2$ in expected polynomial time if the population size stays polynomial with respect to the given inputs. Finally, we experimentally compare the (1+1) EA and GSEMO on different stochastic instances of the NP-hard minimum weight dominating set problem and show that GSEMO significantly outperforms the (1+1) EA for almost all problem instances.

2 Chance Constrained Optimization Problems

Our basic chance-constrained setting is given as follows. Given a set of n items $E = \{e_1, \ldots, e_n\}$ with weights w_i, $1 \leq i \leq n$, we want to solve

$$\min W \quad \text{subject to} \quad Pr(w(x) \leq W) \geq \alpha, \tag{1}$$

where $w(x) = \sum_{i=1}^{n} w_i x_i$, $x \in \{0,1\}^n$, and $\alpha \in [0,1]$. Throughout this paper, we assume that the weights are independent and each w_i is distributed according to a Normal distribution $N(\mu_i, \sigma_i^2)$, $1 \leq i \leq n$, where $\mu_i \geq 1$ and $\sigma_i \geq 1$, $1 \leq i \leq n$. We denote by $\mu_{\max} = \max_{1 \leq i \leq n} \mu_i$ and $v_{\max} = \max_{1 \leq i \leq n} \sigma_i^2$ the maximal expected value and maximal variance, respectively. According to [Ishii *et al.*, 1981], the problem given in Equation 1 is in this case equivalent to minimizing

$$g(x) = \sum_{i=1}^{n} \mu_i x_i + K_\alpha \cdot \left(\sum_{i=1}^{n} \sigma_i^2 x_i \right)^{1/2} \tag{2}$$

where K_α is the α-fractile point of the standard Normal distribution. Throughout this paper, we assume $\alpha \in [1/2, 1[$ as we are interested in solutions of high confidence. Note that there is no finite value of K_α for $\alpha = 1$ due to the infinite tail of the Normal distribution. Our range of α implies $K_\alpha \geq 0$.

We carry out our investigations where there are additional constraints. First, we consider the additional constraint $|x|_1 \geq k$, which requires that at least k items are chosen in each feasible solution. Furthermore, we consider the formulation of the stochastic minimum spanning tree problem given in [Ishii *et al.*, 1981]. Given an undirected connected weighted graph $G = (V, E)$, $n = |V|$ and $m = |E|$ with random weights $w(e_i)$, $e_i \in E$. The search space is $\{0,1\}^m$. For a search point $x \in \{0,1\}^m$, we have $w(x) = \sum_{i=1}^{m} w(e_i) x_i$ as the weight of a solution x. We investigate the problem given in Equation 1 and require for a solution x to be feasible that x encodes a connected graph. We do not require a solution to be a spanning tree in order to be feasible as removing an edge from a cycle in a connected graph automatically improves the solution quality and is being taken care of by the multi-objective algorithms we analyze in this paper. Note that the only difference compared to the previous setting involving the uniform constraint is the requirement that a feasible solution has to be a connected graph.

Algorithm 1: (1+1) EA for minimization

```
Choose x ∈ {0,1}^n uniformly at random;
repeat
    Create y by flipping each bit x_i of x with
      probability 1/n;
    if f(y) ≤ f(x) then
        x ← y;
until stop;
```

3 Analysis of (1+1) EA

The (1+1) EA (Algorithm 1) is a simple evolutionary algorithm using independent bit flips and elitist selection. It is very well studied in the theory of evolutionary computation [Doerr, 2020] and serves as a stepping stone towards the analysis of more complicated evolutionary algorithms. As common, in the area of runtime analysis, we measure the runtime of the (1+1) EA by the number of iterations of the repeat loop. The optimization time refers to the number of fitness evaluations until an optimal solution has been obtained for the first time, and the expected optimization time refers to the expectation of this value.

3.1 Lower Bound for (1+1) EA and Uniform Constraint

We consider the (1+1) EA for the problem stated in Equation 1 with an additional uniform constraint that requires that each feasible solution contains at least k elements, i.e. $|x|_1 \geq k$ holds. We show that the (1+1) EA has an exponential optimization time on an even very restrictive class of instances involving only two different weight distributions.

We use the following fitness function, which should be minimized in the (1+1) EA:

$$f(x) = \begin{cases} g(x) & |x|_1 \geq k \\ (k - |x|_1) \cdot L & |x|_1 < k, \end{cases}$$

where $L = (1 + \sum_{i=1}^{n} \mu_i + K_\alpha (\sum_{i=1}^{n} \sigma_i^2)^{1/2})$. This gives a large penalty to each unit of constraint violation. It implies

n	Result (1+1) EA
100	10/30
200	11/30
500	6/30
1000	3/30
1500	1/30
2000	0/30

Table 1: Success rate for the (1+1) EA on the worst case instance I.

that any feasible solution is better than any infeasible solution and that $f(x) > f(y)$ holds if both x and y are infeasible and $|x|_1 > |y|_1$. Furthermore, the fitness value of an infeasible solution only depends on the number of its elements.

We now show a lower bound on the optimization time of the (1+1) EA for a specific instance class I containing only two types of elements. Type a elements have weights chosen according to $N(n^2 + \delta, 1)$ and type b elements have weights chosen according to $N(n^2, 2)$. We set $\delta = \frac{1}{2\sqrt{k \cdot 1.48}}$. The instance I has exactly $n/2$ elements of type a and $n/2$ elements of type b. We consider $K_\alpha = 1$ which matches $\alpha \approx 0.84134$, and set $k = 0.51n$. Using the fitness function f, we have the additional property for two feasible solutions x and y that $f(x) < f(y)$ if $|x|_1 < |y|_1$ due to an expected weight of at least n^2 for any additional element in a feasible solution. This also implies that an optimal solution has to consist of exactly k elements. The quality of a solution with k elements only depends on the number of type a and type b elements it contains. An optimal solution includes $n/2$ elements of type a whereas a locally optimal solution includes $n/2$ elements of type b. Note that an optimal solution has minimal variance among all feasible solutions. The (1+1) EA produces with high probability the locally optimal solution before the global optimum which leads to the following result.

Theorem 1. *The optimization time of the (1+1) EA on the instance I using the fitness function f is $e^{\Omega(n)}$ with probability $1 - e^{-\Omega(n^{1/4})}$.*

A similar lower bound for a specific class of instances of the chance constrained minimum spanning tree problem having two types of edge weights can be obtained.

We now show the asymptotic behaviour of the (1+1) EA on instance I through an experimental study. Table 1 shows for $n \in \{100, 200, 500, 1000, 1500, 2000\}$ the number of times out of 30 runs the globally optimal solution has been obtained before the locally optimal one. Note that it takes exponential time to escape a locally optimal solution. It can be observed that the fraction of successful runs obtaining the global optimum clearly decreases with n. For $n = 2000$ no globally optimal solution is obtained within 30 runs.

4 Multi-Objective Evolutionary Algorithm

We now introduce bi-objective formulations of the chance constrained problems with uniform and spanning tree constraints. We use a Pareto Optimisation approach for this which computes trade-offs with respect to the expected weight μ and variance v. We say that a solution z dominates a solution x (denoted as $z \preccurlyeq x$) iff $\mu(z) \leq \mu(x)$ and $v(z) \leq v(x)$. We say that z strongly dominates x (donated as $z \prec x$) iff $z \preceq x$ and $\mu(z) < \mu(x)$ or $v(z) < v(x)$.

Algorithm 2: Global SEMO

```
Choose x ∈ {0,1}^n uniformly at random;
P ← {x};
repeat
    Choose x ∈ P uniformly at random;
    Create y by flipping each bit x_i of x with
      probability 1/n;
    if ∄ w ∈ P : w ≺ y then
        S ← (P ∪ {y})\{z ∈ P | y ⪯ z};
until stop;
```

We investigate the algorithm GSEMO [Laumanns *et al.*, 2004; Giel, 2003] shown in Algorithm 2, which has been frequently used in theoretical studies of Pareto optimization. It starts with a solution chosen uniformly at random and keeps at each time step a set of non dominated solutions found so far. In addition to being able to achieve strong theoretical guarantees [Friedrich *et al.*, 2010; Friedrich and Neumann, 2015; Zhou *et al.*, 2019], GSEMO using different types of multi-objective formulation has shown strong performance in practice [Qian *et al.*, 2015; Qian *et al.*, 2017; Roostapour *et al.*, 2022]. We study the multi-objective evolutionary algorithms in terms of the expected time (measured in terms of iterations of the algorithm) until they have produced a population which contains an optimal solution for each $\alpha \in [1/2, 1[$.

4.1 Uniform Constraints

For the case of the uniform constraint $|x|_1 \geq k$, we consider the objective function $f(x) = (\mu(x), v(x))$ where

$$\mu(x) = \begin{cases} \sum_{i=1}^{n} \mu_i x_i & |x|_1 \geq k \\ (k - |x|_1) \cdot (1 + \sum_{i=1}^{n} \mu_i) & |x|_1 < k \end{cases}$$

$$v(x) = \begin{cases} \sum_{i=1}^{n} \sigma_i^2 x_i & |x|_1 \geq k \\ (k - |x|_1) \cdot (1 + \sum_{i=1}^{n} \sigma_i^2) & |x|_1 < k \end{cases}$$

Note that it gives the expected value and variance for any feasible solution, and a large penalty for any unit of constraint violation in each objective function if a solution is infeasible. This implies that the objective value of an infeasible solution is always worse than the value of a feasible solution.

As we have $\mu_i \geq 1$ and $\sigma_i \geq 1$, $1 \leq i \leq n$, each Pareto optimal solution contains exactly k elements. This is due to the fact that we can remove from a solution x with $|x|_1 > k$ any element to obtain a solution y with $\mu(y) < \mu(x)$ and $v(y) < v(x)$. We will minimize $f_\lambda(x) = \lambda\mu(x) + (1-\lambda)v(x)$ by selecting minimal elements with respect to $f_\lambda(e_i) = \lambda\mu_i + (1-\lambda)\sigma_i^2$, $0 < \lambda < 1$. For the special cases $\lambda = 0$ and $\lambda = 1$, we minimize f_λ by minimizing $f_0(x) = (v(x), \mu(x))$ and $f_1(x) = (\mu(x), v(x))$ with respect to the lexicographic order. Note, that we are using f_λ both for the evaluation of a search point x as well as the evaluation of an element e_i. For each fixed $\lambda \in [0, 1]$, an optimal solution for f_λ can be obtained by sorting the items increasing order of f_λ and selecting the first k of them. For a given set X of such points we denote by $X^*_\lambda \subseteq X$ the set of minimal elements with respect to f_λ. Note that all points in the sets X^*_λ, $0 \leq \lambda \leq 1$, are not strongly dominated in X and therefore constitute Pareto optimal points when only considering the set of search points in X.

Definition 1 (Extreme point of set X)**.** *For a given set X, we call $f(x) = (\mu(x), v(x))$ an extreme point of X if there is a $\lambda \in [0, 1]$ such that $x \in X^*_\lambda$ and $v(x) = \max_{y \in X^*_\lambda} v(x)$.*

We denote by $f(X)$ the set of objective vectors corresponding to a set $X \subseteq 2^E$, and by $f(2^E)$ the set of all objective vectors of the considered search space 2^E. The extreme points of 2^E are given by the extreme points of the convex hull of $f(2^E)$. A crucial property of the extreme points is that they contain all objective vectors that are optimal for any $\lambda \in [0, 1]$. Hence, if there is an optimal solution that can be obtained by minimizing f_λ for a (potentially unknown) value of λ, then such a solution is contained in the set of search points corresponding to the extreme points of 2^E.

In the following, we relate an optimal solution of

$$g(x) = \sum_{i=1}^{n} \mu_i x_i + K_\alpha \cdot \left(\sum_{i=1}^{n} \sigma_i^2 x_i \right)^{1/2} \quad \text{subject to } |x|_1 \geq k.$$

to an optimal solution of

$$g_R(x) = R \cdot \sum_{i=1}^{n} \mu_i x_i + K_\alpha \cdot \left(\sum_{i=1}^{n} \sigma_i^2 x_i \right) \quad \text{subject to } |x|_1 \geq k.$$

for a given parameter $R \geq 0$ that determines the weightening of $\mu(x)$ and $v(x)$. Note that g_R is a linear combination of the expected value and the variance and optimizing g_R is equivalent to optimizing f_λ for $\lambda = R/(R + K_\alpha)$ as we have $g_R(x) = (R + K_\alpha) \cdot f_\lambda(x)$ in this case. We use g_R to show that there is a weightening that leads to an optimal solution for g following the notation given in [Ishii *et al.*, 1981], but will work with the normalized weightening of λ when analyzing our multi-objective approach.

Let x^* be an (unknown) optimal solution for g and let $D(x^*) = \left(\sum_{i=1}^n \sigma_i^2 x_i^*\right)^{1/2}$ be its standard deviation. Lemma 1 follows directly from the proof of Theorems 1–3 in [Ishii *et al.*, 1981] where it has been shown to hold for the constraint where a feasible solution has to be a spanning tree. However, the proof only uses the standard deviation of an optimal solution and relates this to the weightening of the expected value and the variance. It therefore holds for the whole search space independently of the constraint that is imposed on it. Therefore, it also holds for the uniform constraint where we require $|x|_1 \geq k$.

Lemma 1 (follows from Theorems 1–3 in [Ishii *et al.*, 1981])**.** *An optimal solution for $g_{2D(X^*)}$ is also optimal for g.*

Based on Lemma 1, an optimal solution for f_λ, where $\lambda = 2D(X^*)/(2D(X^*) + K_\alpha)$, is also optimal for g. As we are dealing with a uniform constraint, an optimal solution for f_λ can be obtained by greedily selecting elements according to f_λ until k elements have been included. The extreme points of the convex hull allow to cover all values of λ where optimal solutions differ as they constitute the values of λ where the optimal greedy solution might change and we bound the number of such extreme points in the following.

In order to identify the extreme points of the Pareto front, we observe that the order of two elements e_i and e_j with respect to a greedy approach selecting always a minimal element with respect to f_λ can only change for one fixed value of λ. We define $\lambda_{i,j} = \frac{\sigma_j^2 - \sigma_i^2}{(\mu_i - \mu_j) + (\sigma_j^2 - \sigma_i^2)}$ for the pair of items e_i and e_j where $\sigma_i^2 < \sigma_j^2$ and $\mu_i > \mu_j$ holds, $1 \leq i < j \leq n$.

Consider the set $\Lambda = \{\lambda_0, \lambda_1, \ldots, \lambda_\ell, \lambda_{\ell+1}\}$ where $\lambda_1, \ldots, \lambda_\ell$ are the values $\lambda_{i,j}$ in increasing order and $\lambda_0 = 0$ and $\lambda_{\ell+1} = 1$. The key observation is that computing Pareto optimal solutions that are optimal solutions for f_λ and every $\lambda \in \Lambda$ gives the extreme points of the problem.

Lemma 2. *Each extreme point of the multi-objective formulation is Pareto optimal and optimal with respect to f_λ for at least one $\lambda \in \Lambda$. The number of extreme points is at most $n(n-1)/2 + 2 \leq n^2$.*

In the following, we assume that $v_{\max} \leq \mu_{\max}$ holds. Otherwise, the bound can be tightened by replacing $v_{\max}$ by $\mu_{\max}$. The following lemma gives an upper on the expected time until GSEMO has obtained Pareto optimal solution of minimal variance. Note that this solution is optimal for f_0. We denote by $P_{\max}$ the maximum population size that GSEMO encounters during the run of the algorithm.

Lemma 3. *The expected time until GSEMO has included a Pareto optimal search point of minimal variance in the population is $O(P_{\max} n^2 (\log n + \log v_{\max}))$.*

Lemma 3 directly gives an upper bound for GSEMO on the worst case instance I for which the (1+1) EA has an exponential optimization time. The feasible solution of minimal variance is optimal for instance I. Furthermore, we have $v_{\max} \leq 2n$ and the variance can only take on $O(n)$ different values. which implies that $P_{\max} = O(n)$ holds for instance I. Therefore, we get the following result for GSEMO on the worst case instance I of the (1+1) EA.

Theorem 2. *The expected time until GSEMO has obtained an optimal solution for the instance I is $O(n^3 \log n)$.*

We now present the result for the general class with a uniform constraint. Based on a Pareto optimal solution having the minimal variance, GSEMO can construct all other extreme points and we obtain the following result.

Theorem 3. *Considering the chance constrained problem with a uniform constraint, the expected time until GSEMO has computed a population which includes an optimal solution for any choice of $\alpha \in [1/2, 1[$ is $O(P_{\max} n^2 \ell (\log n + \log v_{\max}))$.*

Proof. We assume that we have already included a Pareto optimal solution of minimal variance $v_{\min}$ into the population. Let

$$v_\lambda^{\max} = \max_{x \in 2^E} \left\{ v(x) \mid f_\lambda(x) = \min_{z \in 2^E} f_\lambda(z) \right\}$$

and

$$v_\lambda^{\min} = \min_{x \in 2^E} \left\{ v(x) \mid f_\lambda(x) = \min_{z \in 2^E} f_\lambda(z) \right\}$$

be the maximal and minimal variance of any optimal solution for the linear weightening f_λ.

Note that we have $v_\lambda^{\max} = v_\lambda^{\min}$ for $\lambda = 0$ as the Pareto optimal objective vector of minimal variance is unique. Hence, the Pareto optimal solution of minimal variance $v_{\min} = v_0^{\max}$ is a solution of maximal variance for $\lambda = 0$. Consider λ_i, $0 \leq i \leq \ell$. We have $v_{\lambda_i}^{\max} = v_{\lambda_{i+1}}^{\min}$ as the extremal point that is optimal for f_{λ_i} and $f_{\lambda_{i+1}}$ has the largest variance for f_{λ_i} and the smallest variance for $f_{\lambda_{i+1}}$ among the corresponding sets of optimal solutions.

Assume that we have already included into the population a search point x that is minimal with respect to f_{λ_i} and has maximal variance $v_{\lambda_i^{\max}}$ among all these solutions. The solution x is also optimal with respect to $f_{\lambda_{i+1}}$ and we have $v_{\lambda_i}^{\max} = v_{\lambda_{i+1}}^{\min}$. We let r be the number of elements contained in x but not contained in the optimal solution y for $f_{\lambda_{i+1}}$ that has variance $v_{\lambda_{i+1}^{\max}}$. As both solutions contain k elements, differ by r elements and y has maximal variance with respect to $f_{\lambda_{i+1}}$, there are r^2 2-bit flips that bring down the distance $d(x) = v_{\lambda_{i+1}}^{\max} - v(x) \leq v_{\lambda_{i+1}}^{\max} - v_{\lambda_i}^{\max}$. Using the multiplicative drift theorem [Doerr *et al.*, 2012] where we always choose the solution that is optimal with respect $f_{\lambda_{i+1}}$ and has the largest variance, the expected time to reach such as solution of variance $v_{\lambda_{i+1}^{\max}}$ is $O(P_{\max} n^2 \log(v_{\lambda_{i+1}}^{\max} - v_{\lambda_i}^{\max})) = O(P_{\max} n^2 (\log n + \log v_{\max}))$. Summing up over the different values of i, we get $O(P_{\max} n^2 \ell(\log n + \log v_{\max}))$ as an upper bound on the expected time to generate all extreme points. □

4.2 Chance Constrained Minimum Spanning Trees

We now extend the previous results to the chance constrained minimum spanning tree problem where edge weights are independent and chosen according to a Normal distribution. Note that using the expected weight and the variance of a solution as objectives results in a bi-objective minimum spanning tree problems for which a runtime analysis of GSEMO has been provided in [Neumann, 2007].

Let $c(x)$ be the number of connected components of the solution x. We consider the bi-objective formulation for the multi-objective minimum spanning tree problem given in [Neumann, 2007]. Let $w_{ub} = n^2 \cdot \max\{\mu_{\max}, v_{\max}\}$. The fitness of a search point x is given as $f(x) = (\mu(x), v(x))$ where $\mu(x) = (c(x) - 1) \cdot w_{ub} + \sum_{i=1}^{m} \mu_i x_i$ and $v(x) = (c(x) - 1) \cdot w_{ub} + \sum_{i=1}^{m} \sigma_i^2 x_i$. It gives a large penalty for each additional connected component.

We transfer the results for the multi-objective setting under the uniform constraint to the setting where a feasible solution has to be a spanning tree. The crucial observation from [Neumann, 2007] to obtain the extreme points is that edge exchanges resulting in new spanning trees allow to construct solutions on the linear segments between two consecutive extreme points in the same way as in the case of the uniform constraint. Let $\ell \leq m(m-1)/2$ be the pairs of edges e_i and e_j with $\sigma_i^2 < \sigma_j^2$ and $\mu_i > \mu_j$. Similar to Lemma 2 and using the arguments in [Ishii *et al.*, 1981], the number of extreme points is at most $\ell + 2 \leq m^2$ as an optimal solution for $f_\lambda = \lambda\mu(x) + (1 - \lambda)v(x)$ can be obtained by Kruskal's greedy algorithm. We replace the expected time of $O(n^2)$ for an items exchange in the case of the uniform constraint with the expected waiting time of $O(m^2)$ for a specific edge exchange in the case of the multi-objective spanning tree formulation and get the following results.

Theorem 4. *Considering the chance constrained minimum spanning tree problem, the expected time until GSEMO has computed a population which includes an optimal solution for any choice of* $\alpha \in [1/2, 1[$ *is* $O(P_{\max} m^2 \ell(\log n + \log v_{\max}))$.

5 Experimental Investigations

We now present experimental results for the chance constrained version of a classical NP-hard optimization problem. We consider the minimum weight dominating set problem. Given a graph $G = (V, E)$ with weights on the nodes, the goal is to compute a set of nodes D of minimal weight such that each node of the graph is dominated by D, i.e. either contained in D or adjacent to a node in D. Let $n = |V|$ be the number of nodes in the given graph $G = (V, E)$. To generate the benchmarks, we assign each node $u \in V$ a Normal distribution $N(\mu(u), v(u))$ with expected weight $\mu(u)$ and a variance $v(u)$. We consider for each graph values of $\alpha = 1 - \beta$ where $\beta \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}$. The objective function is given by Equation 2 where we consider the chosen nodes together with their weight probability distributions. A solution is feasible if it is a dominating set. Therefore, we use the fitness functions from Section 3.1 for the (1+1) EA and Section 4.1 for GSEMO and have a large penalty term as before for each node in V that is not dominated in the given search point x.

We investigate the graphs cfat200-1, cfat200-2, ca-GrQC and Erdoes992, which are sparse graphs chosen from the network repository [Rossi and Ahmed, 2015]. Each expected weight $\mu(u)$ is an integer chosen independently and uniformly at random in $\{n^2, \ldots, 2n^2\}$ and the variance $v(u)$ is an integer chosen independently and uniformly at random in $\{n^3, \ldots, 2n^3\}$ for cfat200-1 and cfat200-2 and an integer chosen uniformly at random in $\{n, \ldots, n^2\}$ for ca-GrQC and Erdoes992. For the graphs cfat200-1 and cfat200-2 (out of the DIMACS benchmark set), which consist of 200 nodes each, we give each algorithm a budget of $3n^3 \log n$ fitness evaluations. This choice for a relatively small number of nodes is based on common results in the area of runtime analysis. It assumes that the population size is of the order n and that 2-bit flips (costing a factor of n^2) involving a coupon collector effect (factor $\log n$) constitute the essential part of the optimization process. The constant "3" is a bit larger than the usual factor "e" (Eulerian number) often seen in runtime analyses and gives the algorithms a slightly larger budget which would reduce failure probabilities during the run of an algorithm. For the graphs ca-GrQc and Erdos992 (out of the collaboration network benchmark set), which consist of 4158 and 6100 nodes, respectively, we allocate a budget of 10 mil-

Graph	β	(1+1)EA		GSEMO		
		Mean	Std	Mean	Std	p-value
cfat200-1	0.01	**615469**	12021	626202	11513	0.001
	1.0E-4	**636723**	19943	643582	11206	0.015
	1.0E-6	667611	19663	**656385**	10960	0.017
	1.0E-8	681564	19466	**666926**	10780	0.005
	1.0E-10	692730	18762	**676124**	10629	<0.001
cfat200-2	0.01	288254	18040	**276098**	3210	0.088
	1.0E-4	302916	19961	**287393**	3449	0.044
	1.0E-6	314384	20010	**295782**	3634	0.004
	1.0E-8	319220	20135	**302745**	3792	0.003
	1.0E-10	327220	20609	**308822**	3932	0.003
ca-GrQc	0.01	24344710794	411165633	**23236020190**	447391144	<0.001
	1.0E-4	24332158128	466137785	**23236140253**	447391988	<0.001
	1.0E-6	224293904677	387409045	**23236229429**	447392614	<0.001
	1.0E-8	24423330281	408171625	**23236303447**	447393135	<0.001
	1.0E-10	24311919118	394911901	**23236368046**	447393585	<0.001
Erdos992	0.01	83558837836	652555275	**82921436413**	428375728	<0.001
	1.0E-4	83542519561	509659520	**82921667294**	428376235	<0.001
	1.0E-6	83491356567	658566586	**82921838782**	428376612	0.001
	1.0E-8	83569455144	833572201	**82921981120**	428376925	0.001
	1.0E-10	83500515574	585825391	**82922105349**	428377198	<0.001

Table 2: Results for stochastic minimum weight dominating set with different confidence levels of α where $\alpha = 1 - \beta$.

lion fitness evaluations to each run. The goal is here to investigate the quality of the results obtained within a reasonable time budget. GSEMO spends its whole fitness budget on a single run whereas for the (1+1) EA the budget is equally divided among the 5 runs for the different 5 values of α. For each graph and α combination we obtain 30 results as described before. Note that the although GSEMO obtains in one run results for each value of α, the results for a fixed graph and α combination are independent of reach other. We report on the mean and standard deviation of the result obtained for (1+1) EA and GSEMO and use the Mann-Whitney test to compute the p-value in order to examine statistical significance. We call a result statistically significant if the p-value is at most 0.05.

The results are shown in Table 2. It can be observed that GSEMO outperforms the (1+1) EA for 18 out of the 20 settings in terms of the mean value that is achieved in 30 runs. All results are statistically significant. Overall, this shows a clear advantage of using the multi-objective model presented in this paper. We finally report on the maximum population size $P_{\max}$ during the runs as this is an important parameter in the runtime analysis we carried out. Our experiments show that the maximum population is in general small and does not grow exponentially with the number of nodes of the given graph. The maximum population size of GSEMO during any of the 30 runs has been 127 for cfat200-1, 34 for cfat200-2, 15 for ca-GrQc, and 7 for Erdos992. This shows the the maximum population size stays moderate for the problem instances we investigated experimentally.

6 Conclusions

With this paper, we provided the first analysis of evolutionary algorithms for chance constrained combinatorial optimization problems with Normally distributed variables. For the case of uniform constraints we have shown that there are simple instances where the (1+1) EA has an exponential optimization time. Based on these insights we a multi-objective formulation which allows Pareto optimization approaches to compute a set of solutions containing for every possible confidence of α an optimal solution. Finally, we showed the effectiveness of our multi-objective approach by investigating the chance constrained setting of the minimum weight dominating set problem.

Acknowledgments

This work has been supported by the Australian Research Council (ARC) through grant FT200100536 and by the Independent Research Fund Denmark through grant DFF-FNU 8021-00260B.

References

[Assimi *et al.*, 2020] Hirad Assimi, Oscar Harper, Yue Xie, Aneta Neumann, and Frank Neumann. Evolutionary bi-objective optimization for the dynamic chance-constrained knapsack problem based on tail bound objectives. In *ECAI*, volume 325, pages 307–314. IOS Press, 2020.

[Ben-Tal *et al.*, 2009] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[Charnes and Cooper, 1959] Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[Doerr and Neumann, 2020] Benjamin Doerr and Frank Neumann, editors. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer, 2020.

[Doerr *et al.*, 2012] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64:673–697, 2012.

[Doerr *et al.*, 2020] Benjamin Doerr, Carola Doerr, Aneta Neumann, Frank Neumann, and Andrew M. Sutton. Optimization of chance-constrained submodular functions. In *AAAI*, pages 1460–1467. AAAI Press, 2020.

[Doerr, 2020] Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation – Recent Developments in Discrete Optimization*, pages 1–87. Springer, 2020.

[Friedrich and Neumann, 2015] Tobias Friedrich and Frank Neumann. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary Computation*, 23(4):543–558, 2015.

[Friedrich *et al.*, 2010] Tobias Friedrich, Jun He, Nils Hebbinghaus, Frank Neumann, and Carsten Witt. Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4):617–633, 2010.

[Giel, 2003] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *CEC*, volume 3, pages 1918–1925, 2003.

[Ishii *et al.*, 1981] Hiroaki Ishii, Shogo Shiode, Toshio Nishida, and Yoshikazu Namasuya. Stochastic spanning tree problem. *Discret. Appl. Math.*, 3(4):263–273, 1981.

[Jansen, 2013] Thomas Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer, 2013.

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

[Laumanns *et al.*, 2004] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions. *IEEE Trans. Evol. Comput.*, 8(2):170–182, 2004.

[Liu *et al.*, 2013] Bo Liu, Qingfu Zhang, Francisco V. Fernández, and Georges G. E. Gielen. An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization. *IEEE Trans. Evolutionary Computation*, 17(6):786–796, 2013.

[Neumann and Neumann, 2020] Aneta Neumann and Frank Neumann. Optimising monotone chance-constrained submodular functions using evolutionary multi-objective algorithms. In *PPSN (1)*, pages 404–417. Springer, 2020.

[Neumann and Sutton, 2019] Frank Neumann and Andrew M. Sutton. Runtime analysis of the (1 + 1) evolutionary algorithm for the chance-constrained knapsack problem. In *FOGA*, pages 147–153. ACM, 2019.

[Neumann and Wegener, 2006] Frank Neumann and Ingo Wegener. Minimum spanning trees made easier via multi-objective optimization. *Nat. Comput.*, 5(3):305–319, 2006.

[Neumann and Witt, 2010] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.

[Neumann *et al.*, 2022] Aneta Neumann, Yue Xie, and Frank Neumann. Evolutionary algorithms for limiting the effect of uncertainty for the knapsack problem with stochastic profits. *CoRR*, abs/2204.05597, 2022.

[Neumann, 2007] Frank Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *Eur. J. Oper. Res.*, 181(3):1620–1629, 2007.

[Poojari and Varghese, 2008] Chandra A. Poojari and Boby Varghese. Genetic algorithm based technique for solving chance constrained problems. *Eur. J. Oper. Res.*, 185(3):1128–1154, 2008.

[Qian *et al.*, 2015] Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by Pareto optimization. In *NIPS*, pages 1774–1782, 2015.

[Qian *et al.*, 2017] Chao Qian, Jing-Cheng Shi, Yang Yu, and Ke Tang. On subset selection with general cost constraints. In *IJCAI*, pages 2613–2619, 2017.

[Roostapour *et al.*, 2022] Vahid Roostapour, Aneta Neumann, Frank Neumann, and Tobias Friedrich. Pareto optimization for subset selection with dynamic cost constraints. *Artif. Intell.*, 302:103597, 2022.

[Rossi and Ahmed, 2015] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, pages 4292–4293. AAAI Press, 2015.

[Xie *et al.*, 2019] Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, and Frank Neumann. Evolutionary algorithms for the chance-constrained knapsack problem. In *GECCO*, pages 338–346. ACM, 2019.

[Xie *et al.*, 2020] Yue Xie, Aneta Neumann, and Frank Neumann. Specific single- and multi-objective evolutionary algorithms for the chance-constrained knapsack problem. In *GECCO*, pages 271–279. ACM, 2020.

[Xie *et al.*, 2021a] Yue Xie, Aneta Neumann, and Frank Neumann. Heuristic strategies for solving complex interacting stockpile blending problem with chance constraints. In *GECCO*, pages 1079–1087. ACM, 2021.

[Xie *et al.*, 2021b] Yue Xie, Aneta Neumann, Frank Neumann, and Andrew M. Sutton. Runtime analysis of RLS and the (1+1) EA for the chance-constrained knapsack problem with correlated uniform weights. In *GECCO*, pages 1187–1194. ACM, 2021.

[Zhou *et al.*, 2019] Zhi-Hua Zhou, Yang Yu, and Chao Qian. *Evolutionary learning: Advances in theories and algorithms*. Springer, 2019.