Runtime Analysis of the (1+1) Evolutionary Algorithm for the Chance-Constrained Knapsack Problem

Frank Neumann
The University of Adelaide
School of Computer Science
Adelaide, SA, Australia

Andrew M. Sutton
University of Minnesota Duluth
Department of Computer Science
Duluth, MN, USA

ABSTRACT

The area of runtime analysis has made important contributions to the theoretical understanding of evolutionary algoirthms for stochastic problems in recent years. Important real-world applications involve chance constraints where the goal is to optimize a function under the condition that constraints are only violated with a small probability. We rigorously analyze the runtime of the (1+1) EA for the chance-constrained knapsack problem. In this setting, the weights are stochastic, and the objective is to maximize a linear profit function while minimizing the probability of a constraint violation in the total weight. We investigate a number of special cases for this problem, paying attention to how the structure of the chance constraint influences the runtime behavior of the (1+1) EA. Our results reveal that small changes to the profit value can result in hard-to-escape local optima.

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**.

KEYWORDS

Knapsack problem, chance-constrained optimization, evolutionary algorithms

ACM Reference Format:
Frank Neumann and Andrew M. Sutton. 2019. Runtime Analysis of the (1+1) Evolutionary Algorithm for the Chance-Constrained Knapsack Problem. In *Foundations of Genetic Algorithms XV (FOGA '19), August 27–29, 2019, Potsdam, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3299904.3340315

1 INTRODUCTION

The area of runtime analysis has significantly increased the theoretical understanding of evolutionary algorithms and other bio-inspired approaches over the last 30 years [1, 9, 19]. This area of research treats bio-inspired algorithms as a special class of randomized algorithms [20] and analyzes them with respect to their runtime and/or approximation behavior.

In recent years, the analysis of evolutionary algorithms and ant colony optimization for dynamic and stochastic problems has gained significant attention in the literature [4, 14–16, 22]. The survey of Roostapour et al. [23] provides on overview on the results for dynamic and stochastic problems achieved in recent years. In the case of stochastic problems, noise is added to a solution and in most of the cases studied so far in the literature, bio-inspired algorithms are analyzed until they have achieved a solution with a good expected value. However, considering only the expectation does not take the *robustness* of solutions into account, especially if the variance of the value under consideration is high.

An important class of stochastic problems that have been been almost neglected in the area of evolutionary computation are *chance constrained* optimization problems [2, 3]. So far they have received significant attention in the operations research community [6, 7], but to a much lesser extent in the field of evolutionary computation [17]. Chance constrained optimization problems are of very high significance in real-world applications and require that constraints involving stochastic components are only violated with a small probability. Problems that can be modelled using chance constraints therefore include problems with safety requirements in engineering applications, in particular where a failure would involve high costs. Such settings often arise in process optimization [12] and process control [13].

We consider the setting where a deterministic objective function is optimized subject to a stochastic constraint. In its classical form, such a chance constraint requires that it is violated with probability at most α, where α is a parameter that determines the reliability of the solutions with respect to the considered stochastic setting. Recently, single- and multi-objective evolutionary approaches have been proposed for the chance-constrained knapsack problem [26]. These approaches makes use of popular deviation inequalities such as the Chebyshev inequality and Chernoff bounds in order to estimate the constraint violation probability of a given solution. These tools are also frequently used in the area of runtime analysis of evolutionary computation. Studying chance constrained problems in the context of evolutionary algorithms constitutes an important new research direction both from a theoretical and practical perspective.

With this paper, we start the runtime analysis of evolutionary algorithms for chance-constrained problems. Following the design for the chance-constrained knapsack problem introduced in [26], we analyze special cases of this problem. We are particularly interested not just in feasible solutions (which might be active in the chance constraint), but optimal solutions that minimize the probability that the chance constraint is violated. We argue that such solutions are more robust. In terms of theory of evolutionary computation,

FOGA '19, August 27–29, 2019, Potsdam, Germany

ACM ISBN 978-1-4503-6254-2/19/08…$15.00
https://doi.org/10.1145/3299904.3340315

our investigations have the potential of opening up a new area of highly significant research building upon previous classical pseudo-Boolean functions as well as combinatorial optimization problems with deterministic constraints.

In our analysis, we pay special attention to the case in which all the profits are equal. This setting has been investigated in the context of deterministic constrained optimization problems [5]. However, in our case the weights are chosen according to different distributions and we investigate the impact that this has on the search behavior of simple mutation-based evolutionary algorithms. After having investigated the case where all profits are equal to 1, we consider cases where profits and weight distributions differ between the items. We also demonstrate that a very small change to the profit value can result in the presence of local optima that cannot be escaped in expected polynomial time for the investigated algorithm.

The paper is structured as follows. In Section 2, we describe the problem, and the approach for using the Chebyshev inequality to determine constraint violations. Section 3, presents the algorithm and the runtime results for different chance-constraint problems using the Chebyshev inequality for evaluating constraint violations. Finally, we finish with some conclusions

2 PRELIMINARIES

The classical (deterministic) knapsack problem is a constrained combinatorial optimization problem that has been widely studied in the context of evolutionary computation [8, 11, 18, 21, 27]. The objective of the problem is to maximize a linear profit function subject to a linear weight constraint.

In the chance-constrained knapsack problem the weights are stochastic, and the goal is to maximize the profit subject to the probability that the total weight exceeds the given bound is not too high. Formally, the input is given by n items. The stochastic non-negative weights of the items are modeled as n independent random variables $\{w_1, \ldots, w_n\}$, each with expected values $\{a_1, \ldots, a_n\}$ and variances $\{\sigma_1^2, \ldots, \sigma_n^2\}$. Furthermore, we denote by $a_{\max} = \max a_i$ (respectively, $a_{\min} = \min a_i$) the largest (respectively, smallest) expected weight of any item. The non-negative profits of items are deterministic and are denoted by $\{p_1, \ldots, p_n\}$.

A solution is characterized as a vector of 0-1 decision variables $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ where $x_i = 1$ selects the i-th item. The weight of a solution is the random variable

$$W(x) = \sum_{i=1}^{n} w_i x_i$$

with expectation

$$\mathrm{E}[W(x)] = \sum_{i=1}^{n} a_i x_i$$

and variance

$$\mathrm{Var}[W(x)] = \sum_{i=1}^{n} \sigma_i^2 x_i.$$

The chance-constrained knapsack problem can be formulated as follows [10]:

$$\text{maximize} \quad p(x) = \sum_{i=1}^{n} p_i x_i \tag{1}$$

$$\text{subject to} \quad \Pr(W(x) > B) \le \alpha. \tag{2}$$

The objective of this problem is to select a subset of items where the profit is maximized subject to the chance constraint given in Equation (2). The chance constraint requires that a solution violates the constraint bound B with probability at most α.

Theorem 2.1 (One-sided Chebyshev inequality). *Let X be a random variable with finite expectation $\mathrm{E}[X]$ and finite nonzero variance $\mathrm{Var}[X] = \sigma^2$. Then for any $k \in \mathbb{R}^+$,*

$$\Pr(X \ge \mathrm{E}[X] + k) \le \frac{\sigma^2}{\sigma^2 + k^2}.$$

Proof. Set $Y = X - \mathrm{E}[X]$. For any t such that $t + k > 0$,

$$\Pr(X \ge \mathrm{E}[X] + k) = \Pr\left(\frac{Y+t}{k+t} \ge 1\right) \le \Pr\left(\left(\frac{Y+t}{k+t}\right)^2 \ge 1\right). \tag{3}$$

By the Markov inequality, (3) is at most

$$\mathrm{E}\left[\left(\frac{Y+t}{k+t}\right)^2\right] = \frac{\mathrm{E}[Y^2] + 2t\,\mathrm{E}[Y] + t^2}{(k+t)^2} = \frac{\sigma^2 + t^2}{(k+t)^2},$$

since $\mathrm{E}[Y] = 0$ and $\mathrm{E}[Y^2] = \mathrm{E}[(X - \mathrm{E}[X])^2] = \mathrm{E}[X^2] - \mathrm{E}[X] = \sigma^2$. The RHS is minimized by choosing $t = \sigma^2/k$, completing the proof. □

Given a set of stochastic weights with arbitrary distributions, it will be intractable to calculate the exact probability that the chance constraint is violated. Therefore, we will use the one-sided Chebyshev inequality to construct a usable *surrogate* that translates to a guarantee on the feasibility of the chance constraint imposed by Equation (2).

Definition 2.2. For a chance-constrained knapsack instance, we define the surrogate function $\beta\colon \{0,1\}^n \to \mathbb{R}$ over decision vectors as

$$\beta(x) = \frac{\mathrm{Var}[W(x)]}{\mathrm{Var}[W(x)] + (B - \mathrm{E}[W(x)])^2}.$$

It is clear by Theorem 2.1 that $\Pr(W(x) \ge B) \le \beta(x)$, and therefore every x such that $\beta(x) \le \alpha$ is also feasible.

Definition 2.3. Given a solution x with total stochastic weight $W(x)$, we call the difference $B - \mathrm{E}[W(x)]$ the *gap* of x.

It will sometimes be convenient to express feasibility in terms of the gap of a solution. This is formalized as the following lemma, which follows directly from Theorem 2.1.

Lemma 2.4. *If x is a solution vector with gap $k = B - \mathrm{E}[W(x)]$, then the chance constraint stated in Equation (2) is satisfied when $k \ge \sqrt{\mathrm{Var}[W(x)]\frac{1-\alpha}{\alpha}}$, for all $\alpha \in (0,1)$.*

Proof. Setting the gap $k = B - \mathrm{E}[W(x)]$, the LHS of the inequality in the chance constraint (2) can be written as $\Pr(W(x) \ge \mathrm{E}[W(x)] + k)$. By Theorem 2.1, this is bounded above by

$$\frac{\mathrm{Var}[W(x)]}{\mathrm{Var}[W(x)] + k^2} \le \frac{\mathrm{Var}[W(x)]}{\mathrm{Var}[W(x)]\left(1 + \frac{1-\alpha}{\alpha}\right)} = \alpha,$$

Algorithm 1: (1+1) EA for optimizing f [Equation (4)]

Choose x to be a feasible decision vector;
while *stopping criterion not met* **do**
  $y \leftarrow x$;
  foreach $i \in \{1, \ldots, n\}$ **do**
    With probability $1/n$, $y_i \leftarrow (1 - y_i)$;
  if $f(y) \geq f(x)$ **then**
    $x \leftarrow y$

where we have used the claimed lower bound on k. Thus the inequality in (2) is satisfied. □

This lemma provides a useful tool for determining how much gap can ensure that we have a feasible solution and allows for the determination of how much progress the (1+1) EA can make in terms of the given deterministic objective function.

3 RUNTIME ANALYSIS OF THE (1+1) EA

We study the running time of the (1+1) EA defined in Algorithm 1 for optimizing the stochastic-weight knapsack problem under chance constraints. The algorithm starts with an initial feasible solution and produces an offspring y by flipping each bit of of the current solution x with probability $1/n$. The offspring y is accepted if it is not inferior to the parent x.

We are particularly interested in the construction of *robust* optimal solutions, meaning feasible solutions that maximize the profit, but simultaneously minimize the probability that the chance constraint is violated. We define the optimization time of the (1+1) EA as the number of fitness function evaluations necessary until such a solution is constructed. As mentioned previously, we employ the one-sided Chebyshev inequality to bound the probability of violating the chance constraint in order to maintain feasible solutions. In particular, we will study the fitness function

$$f(x) := (p'(x), \beta(x)), \tag{4}$$

where $p'(x) = -1$ iff $\beta(x) > \alpha$ and $p'(x) = p(x)$ otherwise. We optimize f in lexicographic order, where the goal is to maximize $p'(x)$ and minimize $\beta(x)$, i.e. we have

$$f(y) \geq f(x) \Leftrightarrow (p'(y) > p'(x)) \vee ((p'(y) = p'(x)) \wedge (\beta(y) \leq \beta(x)))$$

Note that throughout our investigations we assume that the (1+1) EA is always initialized with a solution that is feasible with respect to the chance constraint. This implies that according to the survival selection, the algorithm does not accept any infeasible solution during the optimization process.

Analyzing the runtime of the (1+1) EA we are interested in the expected number of fitness evaluations until an optimal solution has been produced for the first time. We call this the expected time to achieve an optimal solution and its commonly known as the expected optimization time in the literature.

3.1 Uniform profit

We begin our study with two cases in which the deterministic profits are uniform. In this case, their actual value does not matter, so it is convenient to use unit profits.

3.1.1 *Independent and identically distributed random weights.*

When the stochastic weights are independent and identically distributed, it is easy to see that the behavior of the (1+1) EA is identical to the case of OneMax. Moreover, if the chance constraint parameters given by α and B restricts the value of the optimal solution to any constant fraction of n, then the runtime of the (1+1) EA becomes linear. Consider the following instance.

Instance 1 (Uniform profit, i.i.d weights). *For $1 \leq i \leq n$, let $p_i = 1$, $a_i = a$, for some $a \in \mathbb{R}^+$, and $\sigma_i = c^2$, $c > 0$ a constant.*

Instance 1 resembles a classical OneMax setting where all items have the same property and the goal is to maximize the number of items that are included in the solution.

Theorem 3.1. *If the optimal solution has profit $n - o(n)$, then the (1+1) EA finds an optimal solution to Instance 1 in $O(n \log n)$ time. In the case that the optimal solution has profit at most $n/(1-\epsilon)$ for any constant $0 < \epsilon < 1$, then the expected time until the (1+1) EA finds the optimal solution to Instance 1 is $O(n)$.*

Proof. For any solution $x \in \{0,1\}^n$, we have the bounds

$$\mathrm{E}[W(x)] \leq a \cdot n$$

and

$$\sigma := \sqrt{\mathrm{Var}[W(x)]} \leq c\sqrt{n}.$$

Therefore, in the case the weight bound is

$$B \geq a \cdot n + c\sqrt{n(1-\alpha)/\alpha},$$

then the gap of an arbitrary solution x is

$$\begin{aligned} B - \mathrm{E}[W(x)] \quad &\geq \quad (a \cdot n - \mathrm{E}[W(x)]) + c\sqrt{n}\left(\sqrt{\frac{1-\alpha}{\alpha}}\right) \\ &\geq \quad \sigma\sqrt{\frac{1-\alpha}{\alpha}} \end{aligned}$$

and thus feasible by Lemma 2.4. In such a case, every search point $x \in \{0,1\}^n$ is feasible, and the expected optimization time is $O(n \log n)$.

Otherwise, we argue that every solution $x \in \{0,1\}^n$ such that

$$B - a|x|_1 \geq c\sqrt{|x|_1}\sqrt{\frac{1-\alpha}{\alpha}} \tag{5}$$

is feasible and the others are not feasible via the Chebyshev bound. This follows by Lemma 2.4 and the fact that $\mathrm{Var}[W(x)] := \sigma^2 = c^2|x|_1$.

Rearranging terms in Equation (5), the constraint boundary for Chebyshev-feasible solutions is defined by the set of all $x \in \{0,1\}^n$ such that

$$|x|_1 \leq \max\left\{\frac{\sqrt{4a\left(\frac{1-\alpha}{\alpha}\right)c^2B + \left(\frac{1-\alpha}{\alpha}\right)^2 c^4} + 2aB + \left(\frac{1-\alpha}{\alpha}\right)c^2}{2a^2}, n\right\}.$$

Set $r = \max\{|x|_1 : \beta(x) \leq \alpha\}$ according to the previous expression, and partition $\{0,1\}^n$ by $L_0, L_1, \ldots, L_r$ such that

$$L_i = \{x \in \{0,1\}^n : |x|_1 = i \text{ and } \beta(x) \leq \alpha\}.$$

For all $i < r$, it is possible for the (1+1) EA to generate a feasible solution in L_{i+1} by mutating exactly one of the $n - i$ zero bits to one. This event occurs with probability $(1 - 1/n)^{n-1}(n-i)/n \geq e^{-1}(n-i)/n$. The proof is completed by applying the method of fitness-based partitions to bound the expected optimization time from above as

$$\sum_{i=0}^{r-1} \frac{en}{n-i} = O\Big(n \log\Big(\frac{n}{n-r}\Big)\Big),$$

and the claim follows. □

3.1.2 Two variance classes. Assume now that the stochastic weights are no longer identically distributed. The weights in the following instance all have the same expectation, but there are now two variance classes. Half of the weights share a single low variance value, and the remaining weights share a single high variance value.

Instance 2 (Uniform profits, two variance classes). *For $1 \leq i \leq n$, let $p_i = 1$ and $a_i = a$ for $a \in \mathbb{R}^+$. Let σ_ℓ^2 and σ_h^2 be two positive real numbers with $\sigma_\ell^2 < \sigma_h^2$. For $1 \leq i \leq n/2$, let $\sigma_i^2 = \sigma_\ell^2$, and for $n/2+1 \leq i \leq n$, let $\sigma_i^2 = \sigma_h^2$.*

For Instance 2, the goal is to collect as many items of low variance σ_ℓ^2 and add additional items of high variance σ_h^2 if they still fit into the knapsack.

Theorem 3.2. *The optimization time of the (1+1) EA on Instance 2 is bounded by $O(n^3)$.*

Proof. Let x be an arbitrary feasible decision vector. If

$$\frac{\mathrm{Var}[W(x)] + \sigma_\ell^2}{\mathrm{Var}[W(x)] + \sigma_\ell^2 + (B - (a+1)|x|_1)^2} \leq \alpha, \tag{6}$$

then it is possible to increase the profit by adding a low-variance item. This occurs with probability $\Omega(1/n)$ until either Equation (6) does not hold or all of the low-variance items have been added.

On the other hand, if Equation (6) does not hold, we argue that it is usually possible to reduce the $\beta(x)$ value at a given $|x|_1$ level. Set

$$\ell := \sum_{i=1}^{n/2} x_i, \quad \text{and} \quad h := \sum_{i=n/2+1}^{n} x_i.$$

Then it must be the case that

$$\beta(x) = \frac{\ell\sigma_\ell^2 + h\sigma_h^2}{\ell\sigma_\ell^2 + h\sigma_h^2 + (B - a|x|_1)^2}.$$

Thus, so long as $\ell < n/2$ and $h > 0$, it is possible to reduce $\beta(x)$ without changing $p(x)$ (or the gap $B - a|x|_1$) by swapping a high-variance item for a low-variance item. Such a swap occurs in the (1+1) EA with probability $\Omega(n^{-2})$, and since $\beta(x)$ is reduced, the resulting solution remains feasible.

In the case that $\ell = n/2$, then either (1) a high-variance item can be added (probability $\Omega(1/n)$), or (2) no high-variance items can be added, in which case x is already optimal. The total expected waiting time until an optimal solution is generated is thus bounded by $O(n^3)$, as there are at most n different values for ℓ and h for a given level of $|x|_1$. □

3.2 Dual profit and weight classes

We now turn our attention to the more complicated case where there are two types of items in terms of both profit and weight. The first type of item has a low profit, but the weights are also drawn uniformly from a relatively low interval. The second type of item has a larger profit, but the weights a drawn uniformly from a higher interval.

Instance 3 ($p_i \in \{1, 2\}$, two weight intervals). *Let $p_i = 1$ if $1 \leq i \leq n/2$, otherwise let $p_i = 2$. The stochastic weights are chosen uniformly at random from the real interval $w_i \in [1/2, 3/2]$ for $1 \leq i \leq n/2$, and $w_i \in [3/2, 5/2]$ for $n/2+1 \leq i \leq n$.*

Note that for Instance 3 we have $\sigma_i^2 = 1/12$ for $1 \leq i \leq n$ and $a_i = p_i$ for $1 \leq i \leq n$. The goal is to obtain the highest possible profit by selecting the smallest number of possible items as this leads to a small variance. Hence, items of profit 2 are preferred over items of profit 1.

Theorem 3.3. *The expected optimization time of the (1+1) EA on Instance 3 is $O(n^4)$.*

Proof. Let x be a solution that is non-optimal but feasible and let r be the number of items chosen by x with profit 1 and s be the number of items chosen by x of profit 2. Since $a_i = p_i$ for all $1 \leq i \leq n$, it follows that

$$\mathrm{E}[W(x)] = r + 2s$$

and

$$\mathrm{Var}[W(x)] = (r+s)/12.$$

The probability of violating the chance constraint is bounded by

$$\beta(x) = \frac{(r+s)/12}{(r+s)/12 + (B - (r+2s))^2}$$

If $r < n/2$ and

$$\frac{(r+s+1)/12}{(r+s+1)/12 + (B - (r+2s+1))^2} \leq \alpha, \tag{7}$$

then the profit can be increased by 1 via a 1-bit flip which happens with probability $(n/2 - r)/(en)$.

If $s < n/2$ and

$$\frac{(r+s+1)/12}{(r+s+1)/12 + (B - (r+2(s+1)))^2} \leq \alpha, \tag{8}$$

then the profit can be increased by 2 through a 1-bit flip which happens with probability $(n/2 - s)/(en)$.

If neither condition applies, we argue that it is always possible to decrease r while maintaining the same profit value, and this results in a strictly lower β-value. Consider a fixed profit value $\hat{p}$. All solutions $z \in \{z \mid p(z) = \hat{p}\}$ have the property that

$$\mathrm{E}[W(z)] = p(z)$$

and so the variance only depends on the number of items in solution z. If $r = 1$ or $s = n/2$ and including an additional item of profit 1 would violate the constraint, then x is already optimal. If $r \geq 2$ and $s < n/2$, then removing two profit 1 items and including a profit 2 item, leads to a solution y with $p(y) = p(x)$ and $\beta(y) < \beta(x)$. This solution would be accepted, and contains one fewer low-profit item, reducing the new solution's r-value.

On the other hand, the value of s cannot decrease when accepting solutions of the same profit. If s decreases by any amount, in order to maintain the same profit level, r must increase by twice this amount. This would increase the variance and therefore lead to a solution y with $\beta(y) > \beta(x)$, which would be rejected.

It follows that at any non-optimal point, it is possible to either (1) increase the profit value with probability $\Omega(1/n)$, or (2) decrease the r-value at a profit value level with probability $\Omega(n^{-3})$, resulting in a strictly better β-value.

The β-value can only increase in the case of a strict profit increase. However, we argue that the number of bits necessary to return the β-value to its previous level is proportional to the increase in profit. In particular, define the sequence of nonnegative random variables $\{\delta_t : t \geq 1\}$ to be the profit increase in iteration t.

Suppose x is the current solution in iteration t, and x' is the new solution accepted at the end of iteration t (after line 7 of Algorithm 1). Then the number of surplus 1-bits in x' compared with x is at most $\delta_t/2$. In particular,

$$\beta(x') \leq \frac{\mathrm{Var}[W(x)] + \delta_t/24}{\mathrm{Var}[W(x)] + \delta_t/24 + (B - (\mathrm{E}[W(x)] + \delta_t))^2},$$

where equality holds if there is no change in profit 1 items. The waiting time conditioned on δ_t to reduce the β-value to its previous level by decreasing the r value is at most $c\delta_t n^3$ for some positive constant c.

Denote as T the random variable capturing the optimization time of the (1+1) EA on Instance 3. The total time spent repairing the β-value after profit increases during the run of the (1+1) EA is stochastically dominated by the random sum

$$S_T := \sum_{t=1}^{T} c\delta_t n^3.$$

By Wald's equation [24, 25],

$$\mathrm{E}[S_T] = cn^3\, \mathrm{E}\left[\sum_{t=1}^{T} \mathrm{E}[\delta_t]\right] = O(n^4),$$

since the profit level never decreases during the run, and thus the total expected increase in profit by the end of the run is bounded by $O(n)$. Since an improvement to either objective can be made in every step with probability $\Omega(n^{-3})$ until there are no such improvements possible, the total expected running time is bounded by $O(n^4)$. □

We now change the setting of Instance 3 in a few ways to understand the influence of the instance structure on the runtime behavior of the (1+1) EA. In particular, we see that by adding a small term to the low profit value and explicitly setting the bound and tolerance, it is possible to obtain local optima in the space from which the (1+1) EA does not escape in polynomial time with probability approaching one superpolynomially fast.

Instance 4 ($p_i \in \{1+\epsilon, 2\}$, two weight intervals). *Let $p_i = 1+\epsilon$, $0 < \epsilon < 1$ if $1 \leq i \leq n/2$, otherwise let $p_i = 2$. The stochastic weights are chosen uniformly at random from the real interval $w_i \in [1/2, 3/2]$ for $1 \leq i \leq n/2$, and $w_i \in [3/2, 5/2]$ for $n/2+1 \leq i \leq n$.*

Theorem 3.4. *The search space of Instance 4 with weight bound $B = n/2 + \sqrt{n}$ and confidence level $\alpha = 1/25$ contains local optima from which the (1+1) EA requires superpolynomial time to escape in expectation and with high probability.*

Proof. We prove that the solution $x = (1^{n/2}0^{n/2})$ is locally optimum and difficult to escape. The profit of x is

$$p(x) = \frac{n}{2} + \frac{\epsilon n}{2},$$

with expectation $\mathrm{E}[W(x)] = n/2$ and variance $\mathrm{Var}[W(x)] = n/24$. Furthermore, the chance constraint is *active* for x in the following sense.

$$\beta(x) = \frac{n/24}{n/24 + n} = \frac{1}{25} = \alpha.$$

We argue that in order to construct a distinct feasible solution with profit at least $p(x)$, a large number of decision variables are required to change. Let y be produced from x by removing $\ell < 4\sqrt{n} - 4$ low-profit items and adding $h < \sqrt{n}/2$ high profit items, i.e.,

$$\ell = \sum_{i=1}^{n/2}(x_i - y_i) < 4\sqrt{n} - 4 \quad \text{and} \quad h = \sum_{i=n/2+1}^{n}(y_i - x_i) < \frac{\sqrt{n}}{2}.$$

In order for $p(y) - p(x) \geq 0$, it must be the case that $2h \geq \ell(1+\epsilon) > \ell$.

It will be convenient to define a sequence of analytic functions $g_\ell \colon \mathbb{R} \to \mathbb{R}$ for $\ell \in \{0, 1, \ldots, 4\sqrt{n} - 3\}$ by fixing n.

$$g_\ell(t) = \frac{(n-\ell)/2 + t + 1}{(n-\ell)/2 + t + 1 + 12\left(\sqrt{n} - 2(t+1)\right)^2}. \tag{9}$$

Since

$$\frac{dg_\ell}{dt} = \frac{48\left(\sqrt{n} - 2(t+1)\right)\left((n-\ell)/2 + \sqrt{n} + 2(t+1)\right)}{\left(96\sqrt{n}(t+1) - 25n + \ell - 2(t+1)(48t+49)\right)^2},$$

it follows that g_ℓ is strictly increasing in the interval $t \in \left[0, \frac{\sqrt{n}-2}{2}\right)$. Moreover, since $0 \leq \ell \leq 4\sqrt{n} - 3$,

$$\begin{aligned} g_\ell(0) &= \frac{n - \ell + 2}{25n - 96\sqrt{n} + 98 - \ell} \geq \frac{5 - 4\sqrt{n} + n}{25n - 100\sqrt{n} + 101} \\ &= \frac{1}{25}\left(1 + \frac{24}{25(n - 4\sqrt{n}) + 101}\right) > \frac{1}{25}. \end{aligned}$$

Hence, for any $\ell \in \{0, 1, \ldots, 4\sqrt{n} - 3\}$, $g_\ell(t) > 1/25$ in the interval $t \in \left[0, \frac{\sqrt{n}-2}{2}\right)$.

If y is a solution not identical to x with $p(y) \geq p(x)$ and ℓ and h defined as above, then $\mathrm{Var}[W(y)] = (n/2 - \ell + h)/12$ and $\mathrm{E}[W(y)] = n/2 - \ell + 2h$. Thus,

$$\begin{aligned} \beta(y) &= \frac{(n/2 - \ell + h)}{(n/2 - \ell + h) + 12\left(n/2 + \sqrt{n} - (n/2 - \ell + 2h)\right)^2} \\ &= g_\ell\left(h - \ell/2 - 1\right) > \frac{1}{25}. \end{aligned}$$

The final inequality holds by the fact that the upper bound on $h < \sqrt{n}/2$ guarantees the argument to g_ℓ is strictly less than $(\sqrt{n} - 2)/2$.

We conclude that any solution y constructed from x in which fewer than $4\sqrt{n} - 4$ low-profit items are removed *and* fewer than $\sqrt{n}/2$ high-profit items are removed cannot be feasible. On the other hand, there does exist a solution z constructed from x by removing $\ell = 2\sqrt{n}$ low-profit items and $h = 2\sqrt{n} > \sqrt{n}/2$ high-profit items. This solution has constraint value $\beta(z) = g_{2\sqrt{n}}(\sqrt{n} - 1) = 1/25 = \alpha$. Moreover, the profit of z is

$$p(z) = \frac{n}{2} + \frac{\epsilon n}{2} + 2(1-\epsilon)\sqrt{n} > p(x).$$

Therefore, the (1+1) EA can only escape x by simultaneously changing at least $\Omega(\sqrt{n})$ bits, and the waiting time for an event of this type is geometrically distributed with expectation $n^{\Omega(\sqrt{n})}$. The probability that such an escape occurs in at most n^c steps for an arbitrary constant c is $1 - \left(1 - n^{-\Omega(\sqrt{n})}\right)^{n^c} \leq n^{-\Omega(\sqrt{n})+c}$ by Bernoulli's inequality. □

Finally, we consider an instance where a small term is added to the high profit value.

Instance 5 ($p_i \in \{1, 2+\epsilon\}$, two weight intervals). *Let $p_i = 1$ if $1 \leq i \leq n/2$, otherwise let $p_i = 2+\epsilon$, $0 < \epsilon < 1$. The stochastic weights are chosen uniformly at random from the real interval $w_i \in [1/2, 3/2]$ for $1 \leq i \leq n/2$, and $w_i \in [3/2, 5/2]$ for $n/2 + 1 \leq i \leq n$.*

Theorem 3.5. *The expected optimization time of the (1+1) EA on Instance 5 is $O(n^4)$.*

Proof. Let x be the current feasible solution of the (1+1) EA. If it is possible to add an item without violating the chance constraint, then this can occur with probability at least $\Omega(1/n)$. On the other hand, if no item can be added, we argue that it is always possible to increase the profit by changing two low-profit items for a high-profit item.

Let r be the count of low-profit items in x, and s be the count of high-profit items. Since

$$\beta(x) = \frac{(r+s)/12}{(r+s)/12 + (B - r + 2s)^2} \leq \alpha,$$

if $r > 1$ and $s < n/2$, it is possible to construct a solution y from x by removing two low-profit items and replacing them with a single high-profit item without changing the expected weight, but reducing the variance. This would result in

$$\beta(y) = \frac{(r+s-1)/12}{(r+s-1)/12 + (B - r + 2s)^2} \leq \alpha$$

Hence the profit can always be increased in this way until all $n/2$ high-profit items are included, or until no low-profit items remain. In either case, the solution cannot be improved further, and must be optimal.

A swap of two low-profit items with a single high profit item occurs with probability $\Omega(n^{-3})$. As there are $O(n)$ high-profit items to collect, the proof is complete. □

4 CONCLUSIONS

Chance-constrained optimization problems play a key role in situations where critical stochastic components are involved. Evolutionary algorithms can be used to deal with these problems by using surrogate functions such as the Chebyshev inequality or Chernoff bounds. The main aim of this paper is to introduce the runtime analysis of evolutionary algorithms for chance-constrained problems. This context targets the robustness of solutions to stochastic problems, where we are interested in minimizing the chance that our solution will violate the constraint. We carried out analyses of a number of settings for the chance-constrained knapsack problem, both with uniform profits and instances with varying combinations of dual profit and weight classes. Our proofs are meant to offer insight into the structure of these problems, and to expose where the new challenges lie for deriving runtime bounds in the chance-constrained setting.

Our hope is that the presented work advances a new and challenging area of significant research in the field of runtime analysis for randomized search and optimization heuristics by extending research in combinatorial optimization with deterministic constraints into the direction of stochastic problems.

ACKNOWLEDGEMENTS

This work has been supported by the Australian Research Council through grant DP160102401 and by the South Australian Government through the Research Consortium "Unlocking Complex Resources through Lean Processing".

REFERENCES

[1] Anne Auger and Benjamin Doerr (Eds.). 2011. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific, Singapore. https://doi.org/10.1142/7438

[2] Xiaodi Bai, Xiaojin Zheng, and Xiaoling Sun. 2012. A survey on probabilistically constrained optimization problems. *Numerical Algebra, Control & Optimization* 2, 4 (2012), 767–778. https://doi.org/10.3934/naco.2012.2.767

[3] Abraham Charnes and William W Cooper. 1959. Chance-constrained programming. *Management Science* 6, 1 (1959), 73–79. https://doi.org/10.1287/mnsc.6.1.73

[4] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. 2016. Robustness of Ant Colony Optimization to Noise. *Evolutionary Computation* 24, 2 (2016), 237–254. https://doi.org/10.1162/EVCO_a_00178

[5] Tobias Friedrich, Timo Kötzing, J.A. Gregor Lagodzinski, Frank Neumann, and Martin Schirneck. 2018. Analysis of the (1+1) EA on subclasses of linear functions under uniform and linear constraints. *Theoretical Computer Science* (2018). https://doi.org/10.1016/j.tcs.2018.04.051

[6] Grani Adiwena Hanasusanto, Vladimir Roitch, Daniel Kuhn, and Wolfram Wiesemann. 2015. A distributionally robust perspective on uncertainty quantification and chance constrained programming. *Mathematical Programming* 151, 1 (2015), 35–62. https://doi.org/10.1007/s10107-015-0896-z

[7] Grani Adiwena Hanasusanto, Vladimir Roitch, Daniel Kuhn, and Wolfram Wiesemann. 2017. Ambiguous Joint Chance Constraints Under Mean and Dispersion Information. *Operations Research* 65, 3 (2017), 751–767. https://doi.org/10.1287/opre.2016.1583

[8] Jun He, Boris Mitavskiy, and Yuren Zhou. 2014. A theoretical assessment of solution quality in evolutionary algorithms for the knapsack problem. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 141–148. https://doi.org/10.1109/CEC.2014.6900442

[9] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Springer, Berlin. https://doi.org/10.1007/978-3-642-17339-4

[10] Olivier Klopfenstein and Dritan Nace. 2008. A robust approach to the chance-constrained knapsack problem. *Operations Research Letters* 36, 5 (2008), 628 – 632. https://doi.org/10.1016/j.orl.2008.03.006

[11] Rajeev Kumar and Nilanjan Banerjee. 2005. Running Time Analysis of a Multiobjective Evolutionary Algorithm on Simple and Hard Problems. In *Proceedings of the Eighth Conference on Foundations of Genetic Algorithms (FOGA) (Lecture Notes in Computer Science)*, Vol. 3469. Springer, Berlin, 112–131. https://doi.org/10.1007/11513575_7

[12] Pu Li, Harvey Arellano-Garcia, and Günter Wozny. 2008. Chance constrained programming approach to process optimization under uncertainty. *Computers & Chemical Engineering* 32, 1 (2008), 25 – 45. https://doi.org/10.1016/j.compchemeng.2007.05.009

[13] Pu Li, Moritz Wendt, and Günter Wozny. 2000. Robust model predictive control under chance constraints. *Computers & Chemical Engineering* 24, 2 (2000), 829 – 834. https://doi.org/10.1016/S0098-1354(00)00398-7

[14] Andrei Lissovoi and Carsten Witt. 2015. Runtime analysis of ant colony optimization on dynamic shortest path problems. *Theoretical Computer Science* 561 (2015), 73 – 85. https://doi.org/10.1016/j.tcs.2014.06.035

[15] Andrei Lissovoi and Carsten Witt. 2016. MMAS Versus Population-Based EA on a Family of Dynamic Fitness Functions. *Algorithmica* 75, 3 (01 Jul 2016), 554–576. https://doi.org/10.1007/s00453-015-9975-z

[16] Andrei Lissovoi and Carsten Witt. 2017. A Runtime Analysis of Parallel Evolutionary Algorithms in Dynamic Optimization. *Algorithmica* 78, 2 (01 Jun 2017), 641–659. https://doi.org/10.1007/s00453-016-0262-4

[17] Bo Liu, Qingfu Zhang, Francisco V. Fernández, and Georges G. E. Gielen. 2013. An Efficient Evolutionary Algorithm for Chance-Constrained Bi-Objective Stochastic Optimization. *IEEE Trans. Evolutionary Computation* 17, 6 (2013), 786–796. https://doi.org/10.1109/TEVC.2013.2244898

[18] Frank Neumann and Andrew M. Sutton. 2018. Runtime Analysis of Evolutionary Algorithms for the Knapsack Problem with Favorably Correlated Weights. In *Proceedings of the Fifteenth International Conference on Parallel Problem Solving*

from Nature (Lecture Notes in Computer Science), Anne Auger, Carlos M. Fonseca, Nuno Lourenço, Penousal Machado, Luís Paquete, and Darrell Whitley (Eds.), Vol. 11102. Springer, 141–152. https://doi.org/10.1007/978-3-319-99259-4_12

[19] Frank Neumann and Carsten Witt. 2010. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity* (1st ed.). Springer, Berlin. https://doi.org/10.1007/978-3-642-16544-3

[20] Prabhakar Raghavan and Rajeev Motwani. 1995. *Randomized algorithms*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511814075

[21] Vahid Roostapour, Aneta Neumann, and Frank Neumann. 2018. On the Performance of Baseline Evolutionary Algorithms on the Dynamic Knapsack Problem. In *Proceedings of the Fifteenth International Conference on Parallel Problem Solving from Nature (Lecture Notes in Computer Science)*, Anne Auger, Carlos M. Fonseca, Nuno Lourenço, Penousal Machado, Luís Paquete, and Darrell Whitley (Eds.), Vol. 11101. Springer, Berlin, 158–169. https://doi.org/10.1007/978-3-319-99253-2_13

[22] Vahid Roostapour, Aneta Neumann, Frank Neumann, and Tobias Friedrich. 2018. Pareto Optimization for Subset Selection with Dynamic Cost Constraints. *CoRR* abs/1811.07806 (2018). arXiv:1811.07806 http://arxiv.org/abs/1811.07806 Conference version appears at AAAI 2019.

[23] Vahid Roostapour, Mojgan Pourhassan, and Frank Neumann. 2018. Analysis of Evolutionary Algorithms in Dynamic and Stochastic Environments. *CoRR* abs/1806.08547 (2018). arXiv:1806.08547 http://arxiv.org/abs/1806.08547

[24] Abraham Wald. 1944. On Cumulative Sums of Random Variables. *Ann. Math. Statist.* 15, 3 (09 1944), 283–296. https://doi.org/10.1214/aoms/1177731235

[25] Abraham Wald. 1945. Some Generalizations of the Theory of Cumulative Sums of Random Variables. *Ann. Math. Statist.* 16, 3 (09 1945), 287–293. https://doi.org/10.1214/aoms/1177731092

[26] Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, and Frank Neumann. 2019. Evolutionary Algorithms for the Chance-Constrained Knapsack Problem. *CoRR* abs/1902.04767 (2019). arXiv:1902.04767 http://arxiv.org/abs/1902.04767 Conference version appears at GECCO 2019.

[27] Yuren Zhou and Jun He. 2007. A Runtime Analysis of Evolutionary Algorithms for Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation* 11, 5 (2007), 608–619. https://doi.org/10.1109/TEVC.2006.888929