Evolutionary Algorithms for the Chance-Constrained Knapsack Problem

Yue Xie
University of Adelaide
School of Computer Science
Adelaide, SA, Australia
yue.xie@adelaide.edu.au

Oscar Harper
University of Adelaide
School of Computer Science
Adelaide, SA, Australia
a1704482@student.adelaide.edu.au

Hirad Assimi
University of Adelaide
School of Computer Science
Adelaide, SA, Australia
hirad.assimi@adelaide.edu.au

Aneta Neumann
University of Adelaide
School of Computer Science
Adelaide, SA, Australia
aneta.neumann@adelaide.edu.au

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

ABSTRACT

Evolutionary algorithms have been widely used for a range of stochastic optimization problems. In most studies, the goal is to optimize the expected quality of the solution. Motivated by real-world problems where constraint violations have extremely disruptive effects, we consider a variant of the knapsack problem where the profit is maximized under the constraint that the knapsack capacity bound is violated with a small probability of at most α. This problem is known as chance-constrained knapsack problem and chance-constrained optimization problems have so far gained little attention in the evolutionary computation literature. We show how to use popular deviation inequalities such as Chebyshev's inequality and Chernoff bounds as part of the solution evaluation when tackling these problems by evolutionary algorithms and compare the effectiveness of our algorithms on a wide range of chance-constrained knapsack instances.

CCS CONCEPTS

• **Mathematics of computing → Stochastic control and optimization**;

KEYWORDS

Knapsack problem, chance-constrained optimization, evolutionary algorithms

ACM Reference Format:
Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, and Frank Neumann. 2019. Evolutionary Algorithms for the Chance-Constrained Knapsack Problem. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3321707.3321869

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321869

1 INTRODUCTION

Evolutionary Algorithms (EAs) are bio-inspired randomized optimization techniques and have been shown to be very successful when applied to combinatorial optimization problems. Furthermore, evolutionary algorithms and other bio-inspired computing have been widely applied to various stochastic problems such as stochastic job shop scheduling problem [11], stochastic chemical batch scheduling [34], and other dynamic and stochastic problems [24, 30]. Evolutionary algorithms have the ability to often compute good feasible solutions within a reasonable amount of time, and can easily be applied to stochastic problems. In this paper, we develop evolutionary algorithms for the chance-constrained knapsack problem where the weights of the knapsack are stochastic.

The (deterministic) knapsack problem [12] is one of the best-known NP-hard combinatorial optimization problems. Given a set of n items where each item has a non-negative weight and profit, the goal is to find a selecting of items of maximal profit under the condition that the weight does not exceed a given weight capacity bound B. Different variants of the knapsack problem have been examined in the stochastic setting. In the stochastic knapsack problem, placing each item in the knapsack consumes a random amount of the weight capacity and provides a deterministic profit. It has been shown in [3] that the stochastic knapsack problem is PSPACE-hard. Due to the difficulty of the stochastic knapsack problem, some research are considering the approximation result [1, 3, 25].

Chance-constrained optimization problems [2, 20] have received significant attention in the literature. The main feature of chance constraints is that resulting decision ensures the probability of complying with the constraints, i.e. the confidence level of being feasible. Many previous studies have been made to analyze and efficiently solve chance-constrained optimization problems. Prékopa [27, 28] proposed a dual type algorithm for solving the general chance-constrained problem and discussed several approaches. Hillier [10] proposed a procedure for approximating chance constraints by using linear constraints. Chance constraint programming has been widely applied in different disciplines for optimization under uncertainty [35]. For example, in analog integrated circuit design [18], mechanical engineering [19] and other disciplines [15, 26]. So far, chance constraints have received little attention in the evolutionary computation literature [16].

In this paper, we consider the chance-constrained knapsack problem. The objective is to find a set of items of maximal profit such that the probability that the weight of the selected items exceeds the weight bound B is at most α, where α is a small value limiting the probability of the constraint violation. The chance-constrained knapsack problem has been studied in several published papers. Kleinbery et al. [13] consider only the case where item sizes have a Bernoulli-type distribution (with only two possible sizes for each item). Goel and Indyk [8] proposed an algorithm which violates the chance constraint by a factor of $(1 + \epsilon)$ and provid a PTAS for the case where item sizes have a Poisson or exponential distribution. Vineet Goyal and R. Ravin [9] present a PTAS for the case where item sizes are normally distributed while satisfying the chance constraint strictly by using linear programming.

In our research, the objective is to find a maximum-value set of items such that the probability of the stochastic weights exceeding the capacity bound is at most α. The distribution of weights in our problem use continuous probability distributions like uniform distribution and normal distribution. Furthermore, since EAs easily adapt to stochastic problem, we investigate the performance of evolutionary algorithms for solving the knapsack problem with chance constraints.

In this work, we develop a single-objective evolutionary algorithm approach and a multi-objective evolutionary algorithm approach to solve the chance-constrained knapsack problem. We consider simple single-objective and multi-objective evolutionary algorithms, namely (1+1) EA and Global Simple Evolutionary Multi-objective Optimizer (GSEMO), previously investigated in various studies [7, 21–23, 33] and focus on the formulation of the objective function when establishing the single- and multi-objective model. The main aspect of our work is to estimate the probability that the weight of a given solution exceeds the considered weight bound. Such an estimate is crucial to determine whether a given solution meets the chance constraint and important to guide the search of evolutionary computing techniques.

In order to evaluate a solution with respect to the chance constraint, we use methods commonly used in the area of randomized algorithms [29]. We make use of two inequalities, namely Chebyshev's inequality and Chernoff bound, to calculate an upper bound on the probability of violating the chance constraint. These probabilistic tools allow us to estimate the probability of a constraint violation in a mathematical way without the need for sampling.

To see which of the two inequalities to use in which situation, we carry out an investigation which shows when Chernoff bound is providing tighter bounds dependent on the variance of the given problem. We illustrate the difference of using these two bounds for various confidence levels of α in order to compare the effective of using the two inequalities.

In our experimental investigations, we examine the results obtained by our approaches dependent for a wide range of knapsack instances dependent on confidence level α and the uncertainty of the items. We focus on comparing the results dependent on the use of Chebyshev or Chernoff and on whether to use the single-objective or multi-objective approach. The experimental results shows that if α is small, then the performance of the approaches using Chernoff bound are better than Chebyshev inequality for both algorithms. Furthermore, we observe that the results obtained by GSEMO are usually better than the ones obtained by (1+1) EA.

The remainder of this paper is organized as follows. We introduce the formulation of the problem and the algorithms in Section 2. In Section 3, we present our variant functions basing on the Chebyshev inequality and Chernoff bound. We compare two inequalities in Section 4. Computational experiments and analyses of the obtained results are described in Section 5. Finally, Section 6 concludes the paper.

2 PROBLEM FORMULATION AND ALGORITHMS

In this section, we define the chance-constrained knapsack problem, and present the algorithms considered in this paper.

2.1 Chance-Constrained Knapsack Problem

In the chance-constrained knapsack problem, the input is given by n items with weight and profit and the bound B of the knapsack. The weights of the items are random variables $\{w_1, \ldots, w_n\}$ with expected values $\{a_1, \ldots, a_n\}$ and variances $\{\sigma_1^2, \ldots, \sigma_n^2\}$. The profits of items are deterministic and are denoted by $\{p_1, ..., p_n\}$. The search space of the solution is $\{0, 1\}^n$ which means that item i is chosen **iff** $x_i = 1$, $1 \leq i \leq n$. A solution $X = \{x_1, ..., x_n\} \in \{0, 1\}^n$ has the weight $W(X) = \sum_{i=1}^n w_i x_i$, expected weight $E_W(X) = \sum_{i=1}^n a_i x_i$, variance of weight $Var_W(X) = \sum_{i=1}^n \sigma_i^2 x_i$.

The profit of X is $P(X) = \sum_{i=1}^n p_i x_i$. The chance-constrained knapsack problem can be formulated as follows [14]:

$$\textbf{Maximize} \quad P(X) = \sum_{i=1}^{n} p_i x_i \tag{1}$$

$$\textbf{Subject to} \quad P_r(W(X) \geq B) \leq \alpha \tag{2}$$

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.

For our investigations, we assume that the weights of the items are independent of each other and chosen according to a given distribution which allows to make use of various probabilistic tools.

2.2 Single-Objective Approach

We start by considering a single-objective approach to the chance-constrained knapsack problem and design a fitness function that can be used in single-objective evolutionary algorithms. The fitness function **f** for the approach needs to take into account the chance constraint. We define the fitness of a solution $X \in \{0, 1\}^n$ as:

$$f(X) = (u(X), v(X), P(X)) \tag{3}$$

where $u(X) = max\{E_W(X) - B, 0\}$, $v(X) = max\{P_r(W(X) > B) - \alpha, 0\}$. For this fitness function, $u(X)$ and $v(X)$ need to be minimized and $P(X)$ maximized.

We optimize **f** in lexicographic order and the function **f** takes into account two types of infeasible solutions: (1) the expected weight of the solution exceeding the capacity bound measured by $u(X)$ (2) the probability that the weight of the solution overloading the bound B is bigger than α measured by $v(X)$. Note, that α is usually a small value and throughout this paper we work with

Algorithm 1 (1+1) EA

Choose $x \in \{0,1\}^n$ uniformly at random.
while *stopping criterion not met* **do**
  $y \leftarrow$ flip each bit of x independently with probability of $\frac{1}{n}$;
  if $f(y) \geq f(x)$ **then**
    $x \leftarrow y$;
  end if
end while

$\alpha \leq 0.01$. The reason for using two types of infeasible solutions is that for type (1) we can not use popular tail inequalities such as Chebyshev's inequality of Chernoff bounds to guide the research. However, we can apply them if the expected weight of a solution is below the given capacity bound. Among solutions that meet the chance constraint, we aim to maximize the profit $P(X)$.

Formally, we have

$$f(x) \geq f(y)$$
$$\text{iff } (u(x) \leq u(y)) \vee (u(x) = u(y) \wedge v(x) \leq v(y)) \vee$$
$$(u(x) = u(y) \wedge v(x) = v(y) \wedge P(x) \leq P(y))$$

When comparing two solutions, the feasible solution is preferred in a comparison between an infeasible and feasible solutions. Between two feasible solutions, the one with better profit is preferred. Between two infeasible solutions, the infeasible solution with a lower degree of constraint violation is preferred.

The fitness function **f** can be used in any single-objective evolutionary algorithms. In this paper, we investigate the performance of the classical (1+1) EA (see Algorithm 1) in this paper. The initial solution for the algorithm is a solution with items chosen uniformly at random. (1+1) EA flips each bit of the current solution with the probability of $1/n$ as the mutation step. In the selection step, the algorithm accepts the offspring if it is at least as good as the parent.

2.3 Multi-Objective Approach

Now we consider a multi-objective approach for our problem. In the multi-objective approach, each search point X is a two-dimensional point in the objective space. We use the following fitness function:

$$g_1(X) = \begin{cases} \sum_{i=1}^{n} p_i x_i & g_2(X) \leq \alpha \\ -1 & g_2(X) > \alpha \end{cases} \quad (4)$$

$$g_2(X) = \begin{cases} P_r(W(X) \geq B) & E_W(X) < B \\ 1 + (E_W(X) - B) & E_W(X) \geq B \end{cases} \quad (5)$$

where $W(X)$ denotes the weight of the solution X, $E_W(X)$ denotes the expected weight of solution. We say solution Y dominates solution X w.r.t. g, denoted by $Y \succcurlyeq X$, **iff** $g_1(Y) \geq g_1(X) \wedge g_2(Y) \leq g_2(X)$.

Comparing the two solutions, the objective function g_1 guarantees that a feasible solution dominates every infeasible solutions. Objective function g_2 makes sure that the search process is guided towards feasible solutions and that trade-offs in terms of confidence level and profit are computed for feasible solutions.

The multi-objective algorithm we consider here is the multi-objective evolutionary algorithm (GSEMO) (cf. Algorithm 2) which is inspired from a theoretical study on the performance of evolutionary algorithms in re-optimization under dynamic uniform constraint [31, 32]. It can be seen as a generalization of the (1+1) EA as it uses the same mutation operator and keeps at each time step a set of solutions where each solution is not dominated by any solution found so far in the optimization process. We generate the initially solution by choosing items randomly.

Algorithm 2 GSEMO

Choose $x \in \{0,1\}^n$ uniformly at random
$S \leftarrow \{x\}$;
while stopping criterion not met **do**
  choose $x \in S$ uniformly at random;
  $y \leftarrow$ flip each bit of x independently with probability of $\frac{1}{n}$;
  if ($\nexists w \in S : w \succcurlyeq_{GSEMO} y$) **then**
    $S \leftarrow (S \cup \{y\}) \setminus \{z \in S | y \succcurlyeq_{GSEMO} z\}$;
  end if
end while

3 ESTIMATING CONSTRAINT VIOLATIONS

In this section, we develop the approaches for the fitness functions in Section 2. The approaches are using Chebyshev's inequality and Chernoff bound [29] to calculate the upper bound of chance constraint. Such tools have been widely used in the analysis of evolutionary algorithms [4].

There are two inequality bounds for calculating the upper bound of the chance constraint presented in this section. According to the approaches, (1+1) EA and GSEMO can be used to solve our problem. We consider two different types of intervals for choosing the random weights according to the uniform distribution. The expression of them are $[a_i - \delta, a_i + \delta]$ and $[(1-\beta)a_i, (1+\beta)a_i]$. Here δ and β are parameters that determine the uncertainty of the considered weights.

3.1 Chebyshev's Inequality

The first large deviation bound we consider is Chebyshev's inequality [4]. The inequality has great utility because it can be applied to any probability distribution in which the mean and variance are known. The Chebyshev's inequality is two-sided and considers tails for upper and lower bounds. But in this paper, we only consider the violation of the capacity bound B. Therefore, we use the one-side Chebyshev's inequality which is known as Chebyshev-Cantelli inequality. In order to simplify the presentation, we still use the term Chebyshev inequality in the following.

Theorem 3.1 (Chebyshev inequality). *Let X be a random variable with expectation μ_X and standard deviation σ_X. Then for any $k \in \mathbb{R}^+$,*

$$P(X \geq \mu_X + k) \leq \frac{\sigma_X^2}{\sigma_X^2 + k^2}.$$

For our investigations, we assume that the weight of items are independent. Also let $W(X) = \sum_{i=1}^{n} x_i w_i$ be the weight of a given solution $X = \{x_i, ..., x_n\}$ and let $E_W(X) = \sum_{i=1}^{n} a_i x_i$ be its expected weight derived by linearity of expectation. Furthermore, we can express the variance of weight as $Var_W(X) = \sum_{i=1}^{n} \sigma_i^2 x_i$ as we assume the variables to be independent.

We give a general formula to calculate the probability without identifying the distribution of random variables. Let B be the capacity bound and X be a solution for which we know its expected weight and variance. We get

$$P(W(X) \geq B) \leq \frac{Var_W(X)}{Var_W(X) + (B - E_W(X))^2}. \quad (6)$$

In order to facilitate the readers to select a required inequality according to the existing information, we respectively give the variants of the Chebyshev inequality with a normal distribution and uniform distribution in the two random intervals given in the beginning of this section.

- Uniform distribution with random interval $[a_i - \delta, a_i + \delta]$:

$$P(W(X) \geq B) \leq \frac{\delta^2 \sum_{i=1}^n x_i}{\delta^2 \sum_{i=1}^n x_i + 3(B - E_W(X))^2} \quad (7)$$

- Uniform distribution with random interval $[(1-\beta)a_i, (1+\beta)a_i]$:

$$P(W(X) \geq B) \leq \frac{4\beta^2 (E_W(X))^2}{4\beta^2 (E_W(X))^2 + (2\sqrt{3}B - 2\sqrt{3}E_W(X))^2} \quad (8)$$

- Normal distribution with expectation and variance (a_i, σ_i^2):

$$P(W(X) \geq B) \leq \frac{Var_W(X)}{Var_W(X) + (B - E_W(X))^2} \quad (9)$$

It should be noted that, for a uniform distribution, the expected value and variance can be calculated by considering the interval bounds. For example, a uniform distribution with random interval $[a, b]$, its expected value is $\mu = \frac{a+b}{2}$ and variance is $\sigma^2 = \frac{(b-a)^2}{12}$.

3.2 Chernoff Bound

Chernoff bound provides a sharper tail with exponential decay behaviour. It is a sharper bound than the known tail bounds such as Markov inequality or Chebyshev's inequality, which only yield power-law bounds on tail decay. The Chernoff bound assumes that the variables are independent and take on values in $[0, 1]$.

Theorem 3.2 (Chernoff bound). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0, 1]$. Let $X = \sum_{i=1}^n X_i$ and $\epsilon \geq 0$. Then*

$$P_r[X \geq (1+\epsilon)E(X)] \leq \left(\frac{e^\epsilon}{(1+\epsilon)^{(1+\epsilon)}}\right)^{E(X)}. \quad (10)$$

The theorem is coming from Theorem 10.1 in [4]. There are several versions of Chernoff bound and we the specific one (10.2) in the paper. Theorem 3.2 can be used when all random variables are independent and have their values in $[0, 1]$. Applying this to the chance-constrained knapsack problem, we give a variant of the Chernoff bound to calculate an upper bound for the probability of violating the chance constraint.

Theorem 3.3. *Let the weights of items be independent random variables. Let $w_1, \ldots, w_n$ be the weights of items with expected weight $a_i, \ldots, a_n$ and we assume $w_i \in [a_i - \delta, a_i + \delta]$, $\delta \geq 0$ is the uncertainty in items. Let $B > 0$ be the capacity of the knapsack. Let $W(X) = \sum_{i=1}^n w_i x_i$ for a solution $X = \{x_1, \ldots, x_n\}$ and $E_W(X) = \sum_{i=1}^n a_i x_i$. Then*

$$Pr[W(X) \geq B] \leq \left(\frac{e^{\frac{B - E_W(X)}{\delta \sum_{i=1}^n x_i}}}{\left(\frac{\delta \sum_{i=1}^n x_i + B - E_W(X)}{\delta \sum_{i=1}^n x_i}\right)^{\frac{\delta \sum_{i=1}^n x_i + B - E_W(X)}{\delta \sum_{i=1}^n x_i}}}\right)^{\frac{1}{2}\sum_{i=1}^n x_i} \quad (11)$$

Proof. In order to have random variables in $[0, 1]$ for Chernoff bound, we normalize the weights. In chance-constrained knapsack problem, we assume $w_i \in [a_i - \delta, a_i + \delta]$ and all random variables have the same distance of random interval 2δ but different bounds. Then assume

$$y_i = \frac{w_i - (a_i - \delta)}{2\delta} \in [0, 1], \ Y(X) = \sum_{i=1}^n y_i x_i.$$

Then

$$E_W(y_i) = \frac{a_i - (a_i - \delta)}{2\delta} = \frac{1}{2}$$

and the expected value of the summary of y_i for solution X is

$$E_W[Y(X)] = \frac{E_W(X) - (\sum_{i=1}^n a_i x_i - \sum \delta x_i)}{2\delta} = \frac{1}{2}\sum_{i=1}^n x_i.$$

Using the Chernoff bound of Theorem 3.2, we have

$$\begin{aligned}
&\left(\frac{e^\epsilon}{(1+\epsilon)^{(1+\epsilon)}}\right)^{E_W[Y(X)]} \\
&\geq P_r[Y \geq (1+\epsilon)E_W[Y(X)]] \\
&= Pr[\sum_{i=1}^n \frac{w_i - (a_i - \delta)}{2\delta} x_i \geq (1+\epsilon)\frac{\sum_{i=1}^n x_i}{2}] \\
&= Pr[\sum_{i=1}^n w_i x_i - \sum_{i=1}^n (a_i - \delta) x_i \geq (1+\epsilon)\delta \sum_{i=1}^n x_i] \\
&= Pr[\sum_{i=1}^n w_i x_i \geq \epsilon\delta \sum_{i=1}^n x_i + \sum_{i=1}^n a_i x_i]
\end{aligned}$$

Now, let $B = \epsilon\delta \sum_{i=1}^n x_i + \sum_{i=1}^n a_i x_i$, and get $\epsilon = \frac{B - E_W(X)}{\delta \sum_{i=1}^n x_i}$. Observe that all the above has been proved for any positive real $\delta > a_i$. We now substitute B and ϵ into the last expression which completes the proof. □

There are two main ingredients in the above proof. On the one hand, we studied the random variable Y rather than W. On the other hand, we assume the interval of all weights is same for all weights of the items.

4 COMPARING THE EFFECTIVENESS OF INEQUALITIES

We now examine the relation between Chebyshev's inequality and the Chernoff bound in Theorem 3.2 for our setting. Our goal is to examine under which conditions each is preferred one then other. Let $p_{\text{Cher}}(x)$ be the estimate obtained by Chernoff bound (Theorem 3.2) and $p_{\text{Cheb}}(x)$ be the estimate using Chebyshev's inequality (Theorem 3.1).

The following theorem states a condition on preferring one inequality over the other. Assume that the condition of applying the Chernoff bound and Chebyshev's inequality are sufficient.

Theorem 4.1. *Let X be a solution with expected weight $E_W(X)$ and variance of weight $Var_W(X)$. We have $p_{\mathrm{Cher}}(X) \leq p_{\mathrm{Cheb}}(X)$ if and only if*

$$\frac{\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)} (\epsilon E_W(X))^2}{1-\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)}} \leq Var_W(X) \tag{12}$$

Proof. Using the variable ϵ from the Chernoff bound, we set $k = \epsilon E_W(X)$ in Chebyshev's inequality. Then, we have

$$\begin{aligned}
&\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)} \leq \frac{Var_W(X)}{Var_W(X)+(\epsilon E_W(X))^2} \\
\Leftrightarrow &\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)} (Var_W(X)+(\epsilon E_W(X))^2) \leq Var_W(X) \\
\Leftrightarrow &\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)} (\epsilon E_W(X))^2) \\
&\leq Var_W(X)\left(1-\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)}\right) \\
\Leftrightarrow &\frac{\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)} (\epsilon E_W(X))^2}{1-\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{E_W(X)}} \leq Var_W(X)
\end{aligned}$$

which shows our claim. □

We now investigate this relation a bit further in order to clarify its application. In Theorem 4.1, there are only three parameters, namely ϵ, $E_W(X)$ and $Var_W(X)$, establishing the relation between Chebyshev's inequality and the Chernoff bound. Among them, ϵ indicates the deviation from the expected value. After fixing the value of ϵ, for any problem, the relationship between $E_W(X)$ and $Var_W(X)$ can determine which inequality is more suitable for solving the problem. Figure 1 illustrate for the value of $\epsilon \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$. The figure is based on test problems with 100 items and weights are chosen uniformly at random variables in the interval $[0, 1]$. Every curve in Figure 1 corresponds to a fixed value of ϵ. When the tuple of $[E_W(X), Var_W(X)]$ is located on the curve, then Chernoff bound and Chebyshev's inequality give the same estimate on the probability of a constraint violation. Above the curve, Chernoff bound gives a better estimate, and below the curve Chebyshev's inequality provides a better upper bound on the probability of a constraint violation. As it can be seen from the figure, the greater the deviation of its distribution of weight from the expected weight measured in terms of ϵ, the more suitable the Chernoff bound is for obtaining a superior bound.

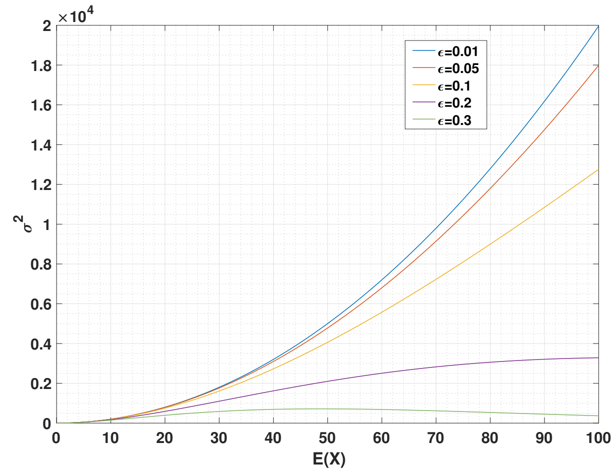

Figure 1: The relationship between $E_W(X)$ and $Var_W(X)$ based on different values of ϵ.

5 EXPERIMENTAL INVESTIGATIONS

In this section, we investigate the performance of the approaches with different fitness functions. We start by describing the experimental setting and the chance-constrained knapsack problem instances. Then we compare the results of the Chebyshev's inequality and Chernoff bound used in the (1+1) EA and GSEMO.

5.1 Experimental Setting

For our experimental investigations, we use benchmarks from [31] which were created following the approach in [17] to obtain different types of instances for the deterministic knapsack problem. There are three different types of instances that are considered. In this paper we only use two of them, namely to *Uncorrelated* and *Bounded Strongly Correlated* instances. In the Uncorrelated (uncorr) instances, the weights and values are integers chosen uniformly at random within $[1, 1000]$. The Bounded Strongly Correlated (bou-s-c) instances are the hardest type of instances and comes from the bounded knapsack problem. The weights of this instance are chosen uniformly at random within $[1, 1000]$ and the profits are set according to the weights within the weight plus a fixed number.

We adapt these instances to the chance-constrained knapsack problem by randomizing the weights. It is essential that the weights of items are positive. To ensure that weights are positive for our desired range of $w_i \in [a_i - \delta, a_i + \delta] \in \mathbb{R}_{\geq 0}$ and $w_i \in [(1-\beta)a_i, (1+\beta)a_i]$, we add a value of γ to the deterministic weight of each item i and take it as its expected weight a_i. As we change the weights of the items, we also need to adjust the considered constraint bound B. However, shifting the knapsack bound is challenging, because it should be assured that a solution remains feasible after the shift in the bound. Moreover, increasing the knapsack capacity expands the feasible search space and may introduce additional feasible solutions. Hence, the shift in knapsack capacity should consider both keeping feasibility of original solutions and size of new feasible search space adaptive. We adjust the original knapsack problem from the benchmark set as follows. First, we sort the weights in

Table 1: Mean value and statistical tests with the uncertainty based on random interval $[a_i - \delta, a_i + \delta]$ and 100 items benchmarks.

	Optimal	α	δ	(1+1) EA						GSEMO			
				Deterministic (1)		Chernoff bound (2)		Chebyshev inequality (3)		Chernoff bound (4)		Chebyshev inequality (5)	
				mean	stat	mean	stat	mean	stat	mean	stat	mean	stat
bou-s-c 1	16047	0.0001	25	15603.20	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	12482.80	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	9793.43	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**12619.43**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	10172.00	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.0001	50	15630.73	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	10944.17	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	6844.60	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**11548.00**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	8132.00	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	15639.00	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	12803.53	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	13380.23	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	12956.50	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**13703.10**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	15681.50	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	11324.87	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	11508.83	$1^{(-)}, 2^{(+)}, 4^{(-)}, 5^{(-)}$	11788.00	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	**11830.70**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(-)}$
		0.01	25	15671.13	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	13203.47	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	14897.73	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	13363.07	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**15249.37**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	15585.70	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	11797.67	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	14141.23	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	12059.33	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**14501.07**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
bou-s-c 2	37926	0.0001	25	37438.70	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	31871.47	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	29334.70	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**32523.30**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	29943.10	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.0001	50	37442.07	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	28996.27	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	23816.70	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**30230.93**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	25270.27	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	37444.20	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	32485.37	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	34541.67	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	33066.67	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**35000.67**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	37478.93	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	29772.00	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	31963.87	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	30785.20	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**32499.37**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	25	37457.57	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	33081.17	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	36522.03	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	33702.83	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**36952.83**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	37481.83	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	30627.67	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	35612.70	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	31403.07	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**36031.97**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
bou-s-c 3	67654	0.0001	25	67117.23	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	58861.13	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	57151.63	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**59735.67**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	57990.90	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.0001	50	67148.43	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	53840.07	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	49535.17	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**54775.37**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	50700.47	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	67152.47	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	59620.63	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	63723.43	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	60542.47	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**64226.40**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	67172.47	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	54943.73	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	60531.63	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	55936.77	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**61125.53**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	25	67113.73	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	60514.23	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	66001.37	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	61425.97	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**66471.80**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	67156.13	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	56523.27	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	64899.50	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	57296.77	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**65438.77**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
uncorr 1	17499	0.0001	25	17244.13	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	12230.23	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	8695.63	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**12343.47**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	8926.00	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.0001	50	17254.40	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	9450.63	$1^{(-)}, 3^{(+)}, 5^{(+)}$	4082.60	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**9552.00**	$1^{(-)}, 3^{(+)}, 5^{(+)}$	4427.00	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	17251.80	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	12718.90	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	14132.73	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	12888.83	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**14344.33**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	17339.40	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	10132.20	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	11385.53	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	10225.00	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**11599.00**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	25	17294.93	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	13397.80	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	16282.37	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	13531.97	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**16472.53**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	17297.50	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	11007.67	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	15299.47	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	11089.70	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**15502.40**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
uncorr 2	29675	0.0001	25	29332.43	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	23173.17	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	21634.37	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**23347.93**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	21910.63	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.0001	50	29377.20	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	19106.03	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	15489.10	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**19306.83**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	15776.47	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	29375.23	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	23839.93	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	26809.73	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	24021.67	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**27018.87**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	29342.40	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	20051.27	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	24344.20	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	20232.43	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**24555.03**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	25	29371.47	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	24581.73	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	28540.70	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	24790.70	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**28798.40**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	29351.30	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	21175.67	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	27712.30	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	21387.20	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**27958.47**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
uncorr 3	41175	0.0001	25	40962.63	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	34567.00	$1^{(-)}, 4^{(-)}, 5^{(-)}$	34539.50	$1^{(-)}, 4^{(-)}, 5^{(-)}$	34682.13	$1^{(-)}, 2^{(+)}, 3^{(+)}$	**34724.93**	$1^{(-)}, 2^{(+)}, 3^{(+)}$
		0.0001	50	40989.97	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	29451.17	$1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$	28055.93	$1^{(-)}, 2^{(-)}, 4^{(-)}, 5^{(-)}$	**29640.93**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 5^{(+)}$	28266.47	$1^{(-)}, 2^{(-)}, 3^{(+)}, 4^{(-)}$
		0.001	25	40961.40	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	35313.40	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	39017.00	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	35440.47	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**39183.30**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.001	50	40928.40	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	30664.20	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	36957.97	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	30811.30	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**37143.10**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	25	40983.03	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	36152.60	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	40379.70	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	36246.23	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**40539.97**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$
		0.01	50	40976.70	$2^{(+)}, 3^{(+)}, 4^{(+)}, 5^{(+)}$	32071.40	$1^{(-)}, 3^{(-)}, 4^{(-)}, 5^{(-)}$	39709.80	$1^{(-)}, 2^{(+)}, 4^{(+)}, 5^{(-)}$	32174.87	$1^{(-)}, 2^{(+)}, 3^{(-)}, 5^{(-)}$	**39918.50**	$1^{(-)}, 2^{(+)}, 3^{(+)}, 4^{(+)}$

the original knapsack instance in ascending order. Then, the first k items with smaller weight are chosen to add in the knapsack till the original capacity is exceeded. Hence, this number of items k represents the most items which any feasible solution may include. We adapt the capacity bound according to this and set

$$B" \leftarrow B + k\gamma. \tag{13}$$

We set $\gamma = 100$ and add that in the initially benchmark. The new benchmarks for the chance-constrained knapsack problem are divided into two categories by the expression of the random intervals like $[a_i - \delta, a_i + \delta]$ and $[(1-\beta)a_i, (1+\beta)a_i]$, where weights are chosen according to the uniform distribution. The tuples (α, δ) and (α, β) are combinations of the elements from the sets of $\alpha = [0.0001, 0.001, 0.01]$, $\delta = [25, 50]$ and $\beta = [0.01, 0.05, 0.1, 0.15]$.

In order to establish a statistical comparison of the results among different probability calculation formulas, we use multiple comparisons tests. For statistical validation, we use the Kruskal-wallis test with 95% confidence for the solutions obtain from both algorithms. Afterwards, we apply the Tukey-Kramer statistical tests that are used for multiple comparison of two or more solutions. For more detailed descriptions on the statistical tests we refer the reader to [5, 6].

5.2 Experimental Results

We benchmark our approach with combinations of the experimental setting described above. We compare the performance of (1+1) EA and GSEMO using Chernoff bound and Chebyshev's inequality separately on instances.

Firstly, we consider the random interval $[a_i - \delta, a_i + \delta]$. Table 1 lists the obtained results for both *bou-s-c* and *uncorr instances*. For each category, we investigate three different instance settings together with different levels of uncertainty determined by δ and the requirement on the chance-constrained determined by α. The column *Optimal* gives the value of the optimum for all instances with deterministic weights by using dynamic programming. For algorithm (1+1) EA, we test the problem with deterministic weights and the two chance-constrained estimation methods: Chebyshev's inequality and Chernoff bound. When the value of α and δ change, they don't infect the result of determined setting. For algorithm GSEMO, we test the problem with the two estimation methods for constrained violation. We obtain the results from 30 independent runs, with 100000 fitness evaluations for each algorithm on each instance. The column *mean* denotes the mean value of 30 runs. Column *stat* denotes the result of statistical testing among the algorithms combined with the estimation methods. The numbers in column *stat* show the significance of the results for each algorithm and constraint violation estimation method. For example, the numbers ($1^{(-)}, 3^{(+)}, 4^{(-)}, 5^{(+)}$) listed in the column *stat* in the first row under *(1+1) EA - Chernoff bound (2)* means that the current one is significantly worse then (1+1) EA for the deterministic setting (1), significantly better than the solutions obtained by *Chebyshev inequality (3), Chebyshev inequality (5)* and significantly worse than

Table 2: Mean value and statistical tests with the uncertainty based on random interval $[(1-\beta)a_i, (1+\beta)a_i]$ and 100 items benchmarks.

	Optimal	α	β	(1+1) EA				GSEMO	
				Deterministic (1)		Chebyshev inequality (2)		Chebyshev inequality (3)	
				mean	stat	mean	stat	mean	stat
bou-s-c (1)	16047	0.001	0.01	15659.33	$2^{(+)}, 3^{(+)}$	15035.90	$1^{(-)}, 3^{(-)}$	**15300.53**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	15653.77	$2^{(+)}, 3^{(+)}$	12920.17	$1^{(-)}, 3^{(-)}$	**13186.33**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	15617.17	$2^{(+)}, 3^{(+)}$	11044.90	$1^{(-)}, 3^{(-)}$	**11234.87**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	15598.27	$2^{(+)}, 3^{(+)}$	9602.30	$1^{(-)}, 3^{(-)}$	**9835.70**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	15582.70	$2^{(+)}, 3^{(+)}$	15389.87	$1^{(-)}, 3^{(-)}$	**15769.43**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	15630.80	$2^{(+)}, 3^{(+)}$	14698.77	$1^{(-)}, 3^{(-)}$	**14956.47**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	15628.83	$2^{(+)}, 3^{(+)}$	13816.00	$1^{(-)}, 3^{(-)}$	**14085.53**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	15627.13	$2^{(+)}, 3^{(+)}$	13044.37	$1^{(-)}, 3^{(-)}$	**13307.53**	$1^{(-)}, 2^{(+)}$
bou-s-c(2)	37926	0.001	0.01	37435.00	$2^{(+)}, 3^{(+)}$	36192.60	$1^{(-)}, 3^{(-)}$	**36567.87**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	37441.07	$2^{(+)}, 3^{(+)}$	32286.53	$1^{(-)}, 3^{(-)}$	**32643.63**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	37409.50	$2^{(+)}, 3^{(+)}$	28538.23	$1^{(-)}, 3^{(-)}$	**28771.57**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	37433.37	$2^{(+)}, 3^{(+)}$	25350.87	$1^{(-)}, 3^{(-)}$	**25676.80**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	37489.23	$2^{(+)}, 3^{(+)}$	37030.50	$1^{(-)}, 3^{(-)}$	**37363.57**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	37433.23	$2^{(+)}, 3^{(+)}$	35557.67	$1^{(-)}, 3^{(-)}$	**35927.80**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	37434.90	$2^{(+)}, 3^{(+)}$	34029.97	$1^{(-)}, 3^{(-)}$	**34313.30**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	37394.50	$2^{(+)}, 3^{(+)}$	32589.43	$1^{(-)}, 3^{(-)}$	**32878.20**	$1^{(-)}, 2^{(+)}$
bou-s-c(3)	67654	0.001	0.01	67154.17	$2^{(+)}, 3^{(+)}$	65163.50	$1^{(-)}, 3^{(-)}$	**65438.53**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	67137.73	$2^{(+)}, 3^{(+)}$	58714.77	$1^{(-)}, 3^{(-)}$	**59231.10**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	67119.23	$2^{(+)}, 3^{(+)}$	52660.57	$1^{(-)}, 3^{(-)}$	**53258.50**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	67169.73	$2^{(+)}, 3^{(+)}$	47840.13	$1^{(-)}, 3^{(-)}$	**48406.60**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	67126.03	$2^{(+)}, 3^{(+)}$	66477.67	$1^{(-)}, 3^{(-)}$	**66753.50**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	67109.57	$2^{(+)}, 3^{(+)}$	64093.33	$1^{(-)}, 3^{(-)}$	**64454.47**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	67149.57	$2^{(+)}, 3^{(+)}$	61537.13	$1^{(-)}, 3^{(-)}$	**61890.33**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	67099.77	$2^{(+)}, 3^{(+)}$	59093.90	$1^{(-)}, 3^{(-)}$	**59591.30**	$1^{(-)}, 2^{(+)}$
uncorr(1)	17499	0.001	0.01	17249.83	$2^{(+)}, 3^{(+)}$	16778.37	$1^{(-)}, 3^{(-)}$	**17052.63**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	17213.60	$2^{(+)}, 3^{(+)}$	15147.57	$1^{(-)}, 3^{(-)}$	**15381.47**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	17290.70	$2^{(+)}, 3^{(+)}$	13393.37	$1^{(-)}, 3^{(-)}$	**13589.00**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	17169.10	$2^{(+)}, 3^{(+)}$	11898.37	$1^{(-)}, 3^{(-)}$	**12209.00**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	17268.83	$2^{(+)}, 3^{(+)}$	17152.97	$1^{(-)}, 3^{(-)}$	**17363.27**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	17322.90	$2^{(+)}, 3^{(+)}$	16647.10	$1^{(-)}, 3^{(-)}$	**16875.00**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	17248.70	$2^{(+)}, 3^{(+)}$	15912.57	$1^{(-)}, 3^{(-)}$	**16141.10**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	17287.00	$2^{(+)}, 3^{(+)}$	15233.23	$1^{(-)}, 3^{(-)}$	**15476.77**	$1^{(-)}, 2^{(+)}$
uncorr(2)	29675	0.001	0.01	29365.87	$2^{(+)}, 3^{(+)}$	28803.67	$1^{(-)}, 3^{(-)}$	**29088.37**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	29353.80	$2^{(+)}, 3^{(+)}$	26758.40	$1^{(-)}, 3^{(-)}$	**27084.83**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	29387.33	$2^{(+)}, 3^{(+)}$	24691.77	$1^{(-)}, 3^{(-)}$	**25033.27**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	29385.80	$2^{(+)}, 3^{(+)}$	22994.10	$1^{(-)}, 3^{(-)}$	**23380.47**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	29419.83	$2^{(+)}, 3^{(+)}$	29209.27	$1^{(-)}, 3^{(-)}$	**29449.90**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	29368.80	$2^{(+)}, 3^{(+)}$	28502.90	$1^{(-)}, 3^{(-)}$	**28779.80**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	29329.00	$2^{(+)}, 3^{(+)}$	27681.83	$1^{(-)}, 3^{(-)}$	**27976.70**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	29344.93	$2^{(+)}, 3^{(+)}$	26889.03	$1^{(-)}, 3^{(-)}$	**27216.10**	$1^{(-)}, 2^{(+)}$
uncorr(3)	41175	0.001	0.01	40976.07	$2^{(+)}, 3^{(+)}$	40472.43	$1^{(-)}, 3^{(-)}$	**40641.67**	$1^{(-)}, 2^{(+)}$
		0.001	0.05	40947.90	$2^{(+)}, 3^{(+)}$	38530.20	$1^{(-)}, 3^{(-)}$	**38717.73**	$1^{(-)}, 2^{(+)}$
		0.001	0.1	40975.67	$2^{(+)}, 3^{(+)}$	36172.50	$1^{(-)}, 3^{(-)}$	**36363.50**	$1^{(-)}, 2^{(+)}$
		0.001	0.15	40976.33	$2^{(+)}, 3^{(+)}$	33878.30	$1^{(-)}, 3^{(-)}$	**34212.23**	$1^{(-)}, 2^{(+)}$
		0.01	0.01	40972.70	$2^{(+)}, 3^{(+)}$	40806.40	$1^{(-)}, 3^{(-)}$	**40964.47**	$1^{(-)}, 2^{(+)}$
		0.01	0.05	40975.97	$2^{(+)}, 3^{(+)}$	40209.10	$1^{(-)}, 3^{(-)}$	**40399.93**	$1^{(-)}, 2^{(+)}$
		0.01	0.1	40945.90	$2^{(+)}, 3^{(+)}$	39423.93	$1^{(-)}, 3^{(-)}$	**39604.53**	$1^{(-)}, 2^{(+)}$
		0.01	0.15	40980.37	$2^{(+)}, 3^{(+)}$	38673.23	$1^{(-)}, 3^{(-)}$	**38863.03**	$1^{(-)}, 2^{(+)}$

Chernoff bound (4). We show the best result among the last four algorithms approaches in bold such that it can be easily observed which approach achieves the best result on a considered instance of the chance-constrained knapsack problem.

The first insight into Table 1 can be draw from the values of column *mean* under *Chernoff bound (2), Chebyshev inequality (3), Chernoff bound (4) and Chebyshev inequality (5)*. We can clearly see that the value for each instance decreases when the value of α decreases with the same δ and when the value of δ increases with the same α. When the uncertainty of the weight is fixed, how the chance-constrained bound α affects the solution quality is shown in Figure 2. The bar chart shows the solutions for instance *bou-s-c 1.* There are four categories in the figure corresponding to the combination of algorithms and inequalities. The three columns of each category correspond to the value of $\alpha \in \{0.0001, 0.001, 0.01\}$, from left to right. We can see from the figure that the mean profit of solutions increases as the chance-constrained bound α is increased. Intuitively, this makes sense as a relaxed requirement on α allows the algorithms to compute solutions that are closer to the bound B and therefore increase their profit.

We now compare the performance of the two algorithms (1+1) EA and GSEMO. By observing the column *stat* in Table 1, we can see when using Chernoff bound to calculate the chance-constrained, GSEMO is always significantly better than (1+1) EA for all instances. Similarly when using Chebyshev inequality to calculate the estimate of a constrained violation, the performance of GSEMO is significantly better than (1+1) EA for all instances. In summary, GSEMO outperforms the (1+1) EA for both elevation inequalities.

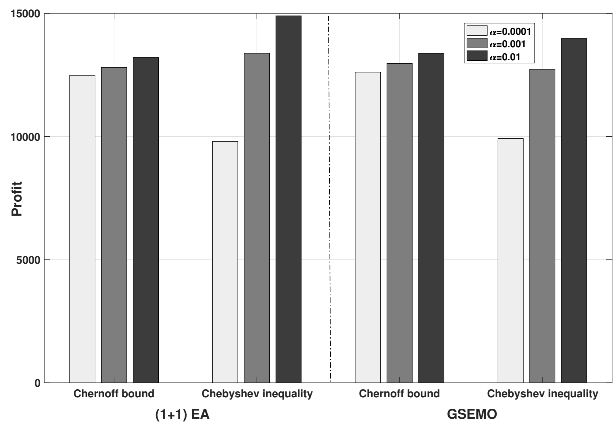

Figure 2: Comparison for different values of α.

We can also observe that for both algorithms, when the value of α is set to be a small number, for example $\alpha = 0.001, 0.01$, the profits obtained using Chernoff bound are mostly significantly better than the one obtained by Chebyshev's inequality. Furthermore, when the value of α is 0.1, the solutions obtained by Chebyshev's inequality are significantly better than the ones obtained by Chernoff bound. The experimental result match to the theoretical implications of Theorem 4.1 and the discussion of this theorem.

We also consider the random interval $[(1-\beta)a_i, (1+\beta)a_i]$. Here, the uncertainty value is expressed as a percentage of the expected value. Since, in this setting, the distance between the minimum and maximum value of the random interval of each item weight is different, the upper bound of the chance constraint cannot be calculated by Chernoff bound and we only use Chebyshev's inequality. The results are shown in Table 2 and we mark in bold the best solutions between (1+1) EA and GSEMO for all instances in every row of the table.

In Table 2 column *Optimal* shows the optimal solution obtained with dynamic programming and column *mean* denotes the mean value for 30 runs, column *stat* gives the statistic test results.

When the uncertainty added to the weight, measured by β, increases, or the upper bound on the chance-constrained in form of α decreases, both algorithms obtain inferior solutions. We compare the solutions of these estimated methods. We can see that the performance of GSEMO is always significantly better than the one of (1+1) EA.

6 CONCLUSIONS

Chance-constrained optimization problems play an important role in various real-world applications and limit the probability of violating a given constraint when dealing with stochastic problems. We have considered the chance-constrained knapsack problem and shown how to incorporate popular probability tail inequalities into the search process of an evolutionary algorithm. Our investigations point out circumstances where to favor Chebyshev inequality or Chernoff bound as part of the fitness evaluation when dealing with the chance-constrained knapsack problem. Furthermore, we have shown that using a multi-objective approach when dealing with the chance-constrained knapsack problem provides a clear benefit compared to its single-objective formulation.

ACKNOWLEDGMENTS

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] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. 2011. *Improved approximation results for stochastic knapsack problems.* 1647–1665.
[2] Abraham Charnes and William W Cooper. 1959. Chance-constrained programming. *Management science* 6, 1 (1959), 73–79.
[3] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. 2008. Approximating the stochastic knapsack problem: the Benefit of adaptivity. *Mathematics of Operations Research* 33, 4 (2008), 945–964.
[4] Benjamin Doerr. 2018. Probabilistic tools for the analysis of randomized optimization heuristics. *arXiv:1801.06733* (2018).
[5] Wade C. Driscoll. 1996. Robustness of the ANOVA and Tukey-Kramer statistical tests. In *International Conference on Computers and Industrial Engineering, CIE 1996*. Pergamon Press, Inc., 265–268.
[6] Asghar Ghasemi and Saleh Zahediasl. 2012. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism* 10, 2 (2012), 486.
[7] Oliver Giel and Ingo Wegener. 2003. Evolutionary algorithms and the maximum matching problem. In *Symposium on Theoretical Aspects of Computer Science, STACS 2003 (Lecture Notes in Computer Science)*, Vol. 2607. Springer, 415–426.
[8] Ashish Goel and Piotr Indyk. 1999. Stochastic load balancing and related problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*. IEEE, 579–586. https://doi.org/10.1109/SFFCS.1999.814632
[9] Vineet Goyal and R. Ravi. 2010. A PTAS for the chance-constrained knapsack problem with random item sizes. *Operations Research Letters* 38, 3 (2010), 161 – 164.
[10] Fredrick S. Hillier. 1967. Chance-constrained programming with 0-1 or bounded bontinuous decision variables. *Management Science* 14, 1 (1967), 34–57.
[11] Shih-Cheng Horng, Shieh-Shing Lin, and Feng-Yi Yang. 2012. Evolutionary Algorithm for Stochastic Job Shop Scheduling with Random Processing Time. *Expert Systems with Applications* 39, 3 (2012), 3603–3610.
[12] Hans Kellerer, Ulrich Pferschy, and David Pisinger. 2005. Knapsack problems. *Business Economics* 33, 2 (2005), 217–219.
[13] Jon Kleinberg, Yuval Rabani, and Éva Tardos. 1997. Allocating bandwidth for bursty connections. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, STOC 1997*. ACM, 664–673.
[14] Olivier Klopfenstein and Dritan Nace. 2008. A robust approach to the chance-constrained knapsack problem. *Operations Research Letters* 36, 5 (2008), 628 – 632.
[15] Baoding Liu. 2007. Uncertainty theory. In *Uncertainty theory*. Springer, 205–234.
[16] 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.
[17] Silvano Martello. 1990. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimiza tion* (1990).
[18] T. McConaghy, P. Palmers, M. Steyaert, and G. G. E. Gielen. 2009. Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28, 9 (2009), 1281–1294.
[19] Lei L Mercado, S-M Kuo, Tien-Yu Lee, and Russell Lee. 2005. Analysis of RF MEMS switch packaging process for yield improvement. *IEEE transactions on advanced packaging* 28, 1 (2005), 134–141.
[20] Bruce L Miller and Harvey M Wagner. 1965. Chance constrained programming with joint constraints. *Operations Research* 13, 6 (1965), 930–945.
[21] Frank Neumann. 2004. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. In *Parallel Problem Solving from Nature - PPSN VIII (Lecture Notes in Computer Science)*, Vol. 3242. Springer, Berlin, Heidelberg, 81–90.
[22] Frank Neumann. 2007. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research* 181, 3 (2007), 1620 – 1629.
[23] Frank Neumann and Ingo Wegener. 2004. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In *Genetic and Evolutionary Computation, GECCO 2004*. ACM, 713–724.

[24] Trung Thanh Nguyen and Xin Yao. 2012. Continuous dynamic constrained optimization - the challenges. *IEEE Transactions on Evolutionary Computation* 16, 6 (2012), 769–786.

[25] Ciara Pike-Burke and Steffen Grunewalder. 2017. Optimistic planning for the stochastic knapsack problem. In *Artificial Intelligence and Statistics*, Vol. 54. PMLR, 1114–1122.

[26] Chandra A Poojari and Boby Varghese. 2008. Genetic algorithm based technique for solving chance constrained problems. *European journal of operational research* 185, 3 (2008), 1128–1154.

[27] András Prékopa. 1990. Dual method for the solution of a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution. *Zeitschrift für Operations Research* 34, 6 (1990), 441–461.

[28] András Prékopa. 1995. Programming under probabilistic constraint and maximizing probabilities under constraints. In *Stochastic Programming*. Springer, 319–371.

[29] Prabhakar Raghavan and Rajeev Motwani. 1995. *Randomized algorithms*. Cambridge University Press Cambridge.

[30] Pratyusha Rakshit, Amit Konar, and Swagatam Das. 2017. Noisy evolutionary optimization algorithms - A comprehensive survey. *Swarm and Evolutionary Computation* 33 (2017), 18–45.

[31] Vahid Roostapour, Aneta Neumann, and Frank Neumann. 2018. On the performance of baseline evolutionary algorithms on the dynamic knapsack problem. In *International Conference on Parallel Problem Solving from Nature - PPSN XV (Lecture Notes in Computer Science)*. Springer, Cham, 158–169.

[32] Feng Shi, Martin Schirneck, Tobias Friedrich, Timo Kötzing, and Frank Neumann. 2017. Reoptimization times of evolutionary algorithms on linear functions under dynamic uniform constraints. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. ACM, 1407–1414.

[33] Feng Shi, Martin Schirneck, Tobias Friedrich, Timo Kötzing, and Frank Neumann. 2018. Reoptimization time analysis of evolutionary algorithms on linear functions under dynamic uniform constraints. *Algorithmica* (2018), 1–30.

[34] Jochen Till, Guido Sand, Maren Urselmann, and Sebastian Engell. 2007. A hybrid evolutionary algorithm for solving two-stage stochastic integer programs in chemical batch scheduling. *Computers & Chemical Engineering* 31, 5-6 (2007), 630–647.

[35] Stanislav Uryasev. 2013. *Probabilistic constrained optimization: methodology and applications*. Vol. 49. Springer Science & Business Media.