Runtime Analysis of Simple Evolutionary Algorithms for the Chance-constrained Makespan Scheduling Problem*

Feng Shi[1], Xiankun Yan[2], and Frank Neumann[2]

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, P.R. China, fengshi@csu.edu.cn
[2] Optimisation and Logistics, School of Computer Science, The University of Adelaide, Adelaide, Australia

Abstract. The Makespan Scheduling problem is an extensively studied NP-hard problem, and its simplest version looks for an allocation approach for a set of jobs with deterministic processing times to two identical machines such that the makespan is minimized. However, in real life scenarios, the actual processing time of each job may be stochastic around an expected value with a variance under the influence of external factors, and these actual processing times may be correlated with covariances. Thus within this paper, we propose a chance-constrained version of the Makespan Scheduling problem and investigate the performance of Randomized Local Search and (1+1) EA for it. More specifically, we study two variants of the Chance-constrained Makespan Scheduling problem and analyze the expected runtime of the two algorithms to obtain an optimal or almost optimal solution to the instances of the two variants.

Keywords: Chance-constraint · Makespan Scheduling problem · RLS · (1+1) EA.

1 Introduction

To discover the reasons behind the successful applications of evolutionary algorithms in various areas including engineering and economics, lots of researchers made efforts to study the theoretical performance of evolutionary algorithms for classical combinatorial optimization problems. But most of these studied problems are deterministic (such as Vertex Cover problem [4,5,10,12,21,22,24,25,26,36] and Minimum Spanning Tree problem [3,11,18,19,31,33]), and the optimization problems in real-world are often stochastic and have dynamic components. Hence in the past few years, the related researchers paid attentions to the theoretical performance of evolutionary algorithms for dynamic and stochastic combinatorial optimization problems [7,13,16,27,28,29] and obtained a series of theoretical results that further advance the understanding of evolutionary algorithms.

* This work has been supported by the National Natural Science Foundation of China under Grants 62072476 and 61872048, the Hunan Provincial Natural Science Foundation of China under Grant 2021JJ40791, and the Australian Research Council (ARC) through grant FT200100536.

Chance-constrained optimization problems is an important class of stochastic optimization problems. They consider that the constraints may be influenced by the noise of stochastic components, thus their goal is to optimize the given objective function under that the constraints can be violated up to certain probability levels [2,9,14,23]. The basic technique for solving chance-constrained optimization problems is to convert the stochastic constraints to their respective deterministic equivalents according to the predetermined confidence level. Recently, researchers began to focus on the *chance-constrained* optimization problems and analyze the theoretical performance of evolutionary algorithms for them.

The classical Makespan Scheduling problem (abbr. MSP) [1] (the simplest version) considers two identical machines and a set of jobs with deterministic processing times, and its aim is to allocate the jobs to the machines such that the makespan is minimized. In real life scenarios, the actual processing time of each job may be stochastic around an expected value with a variance, and the actual processing times of the jobs may be correlated with covariances. Thus a chance-constrained version of MSP, named *Chance-constrained Makespan Scheduling Problem* (abbr. CCMSP), is proposed in the paper. CCMSP considers two identical machines and several groups of jobs, where the jobs have the same expected processing time and variance if they are in the same group, and their actual processing times are correlated by a covariance if they are in the same group and allocated to the same machine. The goal of CCMSP is to minimize a deterministic makespan value and subject to the probability that the actual makespan exceeds the deterministic makespan is no more than an acceptable threshold.

A few theoretical results have been obtained about the performance of evolutionary algorithms for MSP and chance-constrained problems. Witt [32] carried out the runtime analysis of evolutionary algorithms for MSP with two machines. Later Gunia extended the results to MSP with a constant number of machines. Sutton et al. [30] gave the parameterized runtime analysis of RLS and (1+1) EA for MSP with two machines. Neumann et al. [20] proposed the dynamic version of MSP with two machines and analyzed the performance of RLS and (1+1) EA. Xie et al. [34] studied the single- and multi-objective evolutionary algorithms for the Chance-constrained Knapsack problem, where they used the Chebyshev inequality and Chernoff bounds to estimate the constraint violation probability of a given solution. Then Neumann et al. [17] followed the work of Xie et al. [34] and analyzed special cases of this problem. Note that the Chance-constrained Knapsack problem studied in the above two work does not consider the correlationship among the weights of items. Thus recently Xie et al. [35] analyzed the expected optimization time of RLS and (1+1) EA for the Chance-constrained Knapsack Problem with correlated uniform weights. Neumann et al. [15] presented the first runtime analysis of multi-objective evolutionary algorithms for chance-constrained submodular functions.

Within this paper, we investigate the expected runtime of RLS and (1+1) EA for CCMSP. More specifically, we consider two special variants of CCMSP: (1). CCMSP-1, all jobs have the same expected processing time and variance, and all groups have the same *even* size; (2). CCMSP-2, all jobs have the same expected processing time and variance, but the groups have different sizes. For CCMSP-1, we prove that CCMSP-1 is polynomial-time solvable by showing that RLS and (1+1) EA can obtain an optimal solution to any instance I_1 of it in expected runtime $O(n^2/m)$ and $O((k+m)n^2)$, respectively, where n and k are the numbers of jobs and groups considered in I_1, and $m = n/k$. For CCMSP-2, the size difference among groups makes the discussion complicated, thus a simplified variant of CCMSP-2 named CCMSP-2$^+$ is proposed: The sum of the variances and covariances of the jobs allocated to the same machine cannot be over the expected processing time of a job, no matter how many jobs are allocated to the machine. We prove that CCMSP-2$^+$ is NP-hard and that RLS can get an optimal solution to the instance I_2^+ of CCMSP-2$^+$ in expected polynomial-runtime if the total number of jobs is odd; otherwise, an almost optimal solution to I_2^+.

2 Preliminaries

Consider two identical machines M_0 and M_1, and k groups of jobs, where each group G_i has m_i many jobs (i.e., there are $n = \sum_{i=1}^{k} m_i$ many jobs in total). W.l.o.g., assume $m_1 \leq m_2 \leq \ldots \leq m_k$. The j-th job in group G_i ($j \in [1, m_i]$, where the notation $[x, y]$ denotes the set containing all integers ranging from x to y), denoted by b_{ij}, has actual processing time p_{ij} with expect value $E[p_{ij}] = a_{ij} > 0$ and variance $\sigma_{ij}^2 > 0$. Additionally, for any two jobs of the same group G_i, if they are allocated to the same machine, then their actual processing times are correlated with each other by a covariance $c_i > 0$; otherwise, independent.

The *Chance-constrained Makespan Scheduling Problem* (abbr. CCMSP) studied in the paper looks for an allocation of the n jobs to the two machines that minimizes the makespan M such that the probabilities of the *loads* on M_0 and M_1 exceeding M are no more than a threshold $0 < \gamma < 1$, where the load on M_t ($t \in [0, 1]$) is the sum of the actual processing times of the jobs allocated to M_t.

An allocation (or simply called solution) x to an instance of CCMSP, is represented as a bit-string with length n, $x = x_{11} \cdots x_{ij} \cdots x_{km_k} \in \{0, 1\}^n$, where the job b_{ij} is allocated to M_0 if $x_{ij} = 0$; otherwise, M_1 (in the remaining text, we simply say that a bit is of G_i if its corresponding job is of G_i). Denote by $M_0(x)$ and $M_1(x)$ the sets of jobs allocated to M_0 and M_1, respectively, w.r.t. x. Denote by $l_t(x) = \sum_{b_{ij} \in M_t(x)} p_{ij}$ the *load* on M_t ($t \in [0, 1]$). Let $\alpha_i(x) = |M_0(x) \cap G_i|$ and $\beta_i(x) = |M_1(x) \cap G_i|$ for all $i \in [1, k]$. The CCMSP can be formulated as:

$$\begin{aligned} &\textbf{Minimize} \quad M \\ &\textbf{Subject to} \quad \Pr(l_t(x) > M) \leq \gamma \ \text{ for all } \ t \in [0, 1]. \end{aligned}$$

Observe that the excepted value of $l_t(x)$ is $E[l_t(x)] = \sum_{b_{ij} \in M_t(x)} a_{ij}$. Considering the variance σ_{ij}^2 of each job b_{ij} and the covariance among the jobs of the same group that are allocated to the same machine, the variance of $l_t(x)$ is $Var[l_t(x)] = \sum_{b_{ij} \in M_t(x)} \sigma_{ij}^2 + cov[l_t(x)]$, where $cov[l_t(x)] = \sum_{i=1}^{k} 2c_i \binom{|M_t(x) \cap G_i|}{2}$. Note that $\binom{|M_t(x) \cap G_i|}{2} = 0$ if $0 \leq |M_t(x) \cap G_i| \leq 1$. For the probability $Pr(l_t(x) > M)$ with $t \in [0, 1]$, as the work [34,35], we use the one-sided Chebyshev's inequality (cf. Theorem 1) to construct a usable surrogate of the chance-constraint.

Theorem 1. *(One-sided Chebyshev's inequality). Let X be a random variable with expected value $E[X]$ and variance $Var[X]$. Then for any $\Delta \in \mathbb{R}^+$, $\Pr(X > E[X] + \Delta) \leq \frac{Var[X]}{Var[X]+\Delta^2}$.*

By the One-sided Chebyshev's inequality, upper bounding the probability of the actual makespan exceeding M by γ indicates that for all $t \in [0, 1]$,

$$\Pr(l_t(x) > M) \leq \frac{Var[l_t(x)]}{Var[l_t(x)] + (M - E[l_t(x)])^2} \leq \gamma$$
$$\iff \sqrt{\frac{(1-\gamma)}{\gamma} Var[l_t(x)]} + E[l_t(x)] = l'_t(x) \leq M.$$

Thus $\max\{\Pr(l_0(x) > M), \Pr(l_1(x) > M)\} \leq \gamma$ hold iff $L(x) = \max\{l'_0(x), l'_1(x)\} \leq M$. In other words, $L(x)$ is the tight lower bound for the value of M, if using the surrogate of the chance-constraint by the One-sided Chebyshev's inequality. Therefore, $l'_t(x)$ can be treated as a *new* measure for the load on M_t, and the goal of CCMSP is simplified to minimize $L(x)$. Let $t(x) = \arg\max_t\{l'_0(x), l'_1(x)\}$.

It is not hard to derive that CCMSP is NP-hard as MSP is NP-hard. Within the paper, we study the two specific variants of CCMSP given below.

CCMSP-1. All the n jobs have the same expected processing time $a_{ij} = a > 0$ and variance $\sigma_{ij}^2 = d > 0$, and the k groups have the same covariance $c > 0$ and size $m > 0$. Moreover, m is even.

CCMSP-2. All the n jobs have the same expected processing time $a_{ij} = a > 0$ and variance $\sigma_{ij}^2 = d > 0$, and the k groups have the same covariances $c > 0$. However, the k groups may have different sizes (may be even or odd).

Given an instance I of CCMSP-1 or CCMSP-2 and a solution x to I, if $||M_0(x)| - |M_1(x)|| \leq 1$ (i.e., $|M_0(x)| = |M_1(x)|$ if n is even), then x is an *equal-solution*; if $||M_0(x)| - |M_1(x)|| \leq 1$, and $|\alpha_i(x) - \beta_i(x)| \leq 1$ for all $i \in [1, k]$ (i.e., $\alpha_i = \beta_i$ if m_i is even), then x is a *balanced-solution*.

3 Algorithms

We study the performance of Randomized Local Search (abbr. RLS, given as Algorithm 1) and (1+1) EA (given as Algorithm 2) for the two variants of CCMSP. The two algorithms run in a similar way, randomly generating an offspring based on the maintained solution and replacing it if the offspring is not worse than it regarding their fitness. The difference between the two algorithms is the way to generate offspring: With probability $1/2$, RLS chooses one bit of the maintained solution uniformly at random and flips it, and $1/2$ chooses two bits of the maintained solution uniformly at random and flips them; (1+1) EA flips each bit of the maintained solution with probability $1/n$. The fitness function considered in the two algorithms is the natural one, $f(x) = L(x) = \max\{l'_0(x), l'_1(x)\}$.

4 Performance for CCMSP-1

The section starts with an observation that will be used throughout the paper.

Algorithm 1: RLS

choose $x \in \{0,1\}^n$ uniformly at random;
while *stopping criterion not met* **do**
  choose $b \in \{0,1\}$ uniformly at random;
  if $b = 0$ **then**
    $y \leftarrow$ flip one bit of x chosen uniformly at random;
  else
    choose $(i,j) \in \{(k,l) | 1 \leq k < l \leq n\}$ uniformly at random;
    $y \leftarrow$ flip the i-th and j-th bits of x;
  if $f(y) \leq f(x)$ **then**
    $x \leftarrow y$;

Algorithm 2: (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 $1/n$;
  if $f(y) \leq f(x)$ **then**
    $x \leftarrow y$;

Observation 1 $\binom{\lfloor \frac{x+y}{2} \rfloor}{2} + \binom{\lceil \frac{x+y}{2} \rceil}{2} \leq \binom{x}{2} + \binom{y}{2} \leq \binom{x+y}{2}$ *holds for any two natural numbers x and y.*

Consider an instance $I_1 = (a, c, d, \gamma, k, m)$ of CCMSP-1 and a solution x to I_1. As the groups considered in I_1 have the same size m, there is a variable $\delta_i(x)$ such that $\alpha_i(x) = \frac{m}{2} + \delta_i(x)$ and $\beta_i(x) = \frac{m}{2} - \delta_i(x)$ for any $i \in [1, k]$. Thus,

$$\begin{aligned} cov[l_0(x)] - cov[l_1(x)] &= 2c \sum_{i=1}^{k} \left(\binom{\alpha_i}{2} - \binom{\beta_i}{2} \right) = 2c(m-1) \sum_{i=1}^{k} \delta_i \\ &= c(m-1) \left(\sum_{i=1}^{k} \alpha_i(x) - \sum_{i=1}^{k} \beta_i(x) \right) = c(m-1) \left(|M_0(x)| - |M_1(x)| \right). \end{aligned}$$

Based on the conclusion, it is not hard to derive the following two lemmata.

Lemma 1. *For any solution x to the instance $I_1 = (a, c, d, \gamma, k, m)$ of CCMSP-1, if $|M_0(x)| > |M_1(x)|$ (resp., $|M_1(x)| > |M_0(x)|$) then $l'_0(x) > l'_1(x)$ (resp., $l'_1(x) > l'_0(x)$); if $|M_0(x)| = |M_1(x)|$ then $l'_0(x) = l'_1(x)$.*

Lemma 2. *For any solution x to the instance $I_1 = (a, c, d, \gamma, k, m)$ of CCMSP-1, if x is a balanced-solution then $L(x) = l'_0(x) = l'_1(x)$ gets the minimum value; more specifically, x is an optimal solution to I_1 iff x is a balanced-solution to I_1.*

Theorem 2. *The expected runtime of RLS to obtain an optimal solution to the instance $I_1 = (a, c, d, \gamma, k, m)$ of CCMSP-1 is $O(n^2/m) = O(kn)$.*

Proof. Let x_0 be the initial solution maintained by RLS. Assume that $|M_0(x_0)| > |M_1(x_0)|$. Thus $L(x) = l_0'(x_0) > l_1'(x_0)$ by Lemma 1 and $|M_0(x_0)| - |M_1(x_0)| \geq 2$ as $n = mk$ is even. The following discussion first analyzes the process of RLS to obtain the first equal-solution x_1 based on x_0. Five possible cases for the mutation of RLS on x_0 are listed as follows, obtaining an offspring x_0' of x_0.

Case (1). Flipping a 0-bit in x_0 (i.e., $|M_0(x_0')| = |M_0(x_0)| - 1$). Observe that $L(x_0) = l_0'(x_0) > l_0'(x_0')$. As $|M_0(x_0)| - |M_1(x_0)| \geq 2$, $|M_0(x_0')| \geq |M_1(x_0')|$ and $L(x_0') = l_0'(x_0')$ by Lemma 1. Thus $L(x_0') < L(x_0)$ and x_0' can be accepted.

Case (2). Flipping a 1-bit in x_0 (i.e., $|M_0(x_0')| = |M_0(x_0)| + 1$). Observe that $L(x_0') = l_0'(x_0') > l_0'(x_0) = L(x_0)$, thus x_0' cannot be accepted.

Case (3). Flipping two 0-bits in x_0 (i.e., $|M_0(x_0')| = |M_0(x_0)| - 2$). If $|M_0(x_0')| \geq |M_1(x_0')|$, then using the reasoning for Case (1) gets that $L(x_0') \leq L(x_0)$ and x_0' can be accepted. If $|M_0(x_0')| < |M_1(x_0')|$ then $|M_0(x_0)| = |M_1(x_0')|$ as n is even. By Lemma 1, $L(x_0) - L(x_0') = \sqrt{\frac{1-\gamma}{\gamma}} \left(\sqrt{Var[l_0(x_0)]} - \sqrt{Var[l_1(x_0')]} \right)$. As $Var[l_0(x_0)] \geq Var[l_1(x_0')] \iff cov[l_0(x_0)] \geq cov[l_1(x_0')]$, x_0' can be accepted iff $cov[l_0(x_0)] \geq cov[l_1(x_0')]$.

Case (4). Flipping a 0-bit and a 1-bit in x (i.e., $|M_0(x_0')| = |M_0(x_0)|$). Using the reasoning similar to that for Case (3), we have that x_0' can be accepted iff $cov[l_0(x_0)] \geq cov[l_0(x_0')]$. **Case (5)**. Flipping two 1-bits in x_0 (i.e., $|M_0(x_0')| = |M_0(x_0)| + 2$). Using the reasoning similar to that for Case (2), we have that $L(x_0') > L(x_0)$ and x_0' cannot be accepted.

Summarizing the above analysis gets that if x_0' is accepted by RLS, then it satisfies one of the following two conditions: (1). $|M_{t(x_0')}(x_0')| < |M_0(x_0)|$ and $cov[l_{t(x_0')}(x_0')] < cov[l_0(x_0)]$; (2). $|M_{t(x_0')}(x_0')| = |M_0(x_0)|$ and $cov[l_{t(x_0')}(x_0')] \leq cov[l_0(x_0)]$. That is, the gap between the numbers of jobs in the two machines cannot increase during the optimization process. The mutation considered in Case (1) can be generated by RLS with probability $\Omega(1/4)$ that decreases the gap between the numbers of jobs in the two machines by 2. As $||M_0(x_0)| - |M_1(x_0)|| \leq n$, using the Additive Drift analysis [8] gets that RLS takes expected runtime $O(n)$ to obtain the first equal-solution x_1 based on x_0.

Now we consider the expected runtime of RLS to obtain an optimal solution x^* based on x_1. Let $p(x) = \sum_{i=1}^{k} |\alpha_i(x) - \beta_i(x)| = \sum_{i=1}^{k} |2\alpha_i(x) - m|$ be the potential of the solution x maintained during the process, and we show that during the optimization process the potential value cannot increase. Note that once the first equal-solution x_1 is obtained, then all solutions subsequently accepted by RLS are equal-ones, thus only the mutations flipping a 0-bit and a 1-bit of x_1 are considered below. Assume that the mutation flips a 0-bit of G_i and a 1-bit of G_j in x_1, and denoted by x_1' the solution obtained. The potential change is

$$\Delta_p = p(x_1) - p(x_1') = |2\alpha_i(x_1) - m| + |2\alpha_j(x_1) - m| - \left(|2\alpha_i(x_1') - m| + |2\alpha_j(x_1') - m| \right),$$

where $\alpha_i(x_1') = \alpha_i(x_1) - 1$ and $\alpha_j(x_1') = \alpha_j(x_1) + 1$. The above discussion shows that x_1' can be accepted by RLS iff $\Delta_{cov} = cov[l_0(x_1)] - cov[l_0(x_1')] \geq 0$, where

$$\Delta_{cov}/2c = (cov[l_0(x_1)] - cov[l_0(x_1')])/2c = \alpha_i(x_1) - 1 - \alpha_j(x_1).$$

We divide the analysis for the values of Δ_p and Δ_{Var} into four cases.

Case (I). $\alpha_i(x_1) > \frac{m}{2}$ and $\alpha_j(x_1) \geq \frac{m}{2}$. Observe that $\Delta_p = 0$, but the value of Δ_{Var} depends on the relationship between $\alpha_i(x_1)$ and $\alpha_j(x_1)$.

Case (II). $\alpha_i(x_1) \leq \frac{m}{2}$ and $\alpha_j(x_1) \geq \frac{m}{2}$. Observe that $\Delta_p = -4$, but $\Delta_{Var} < 0$, implying that x_1' cannot be accepted by RLS.

Case (III). $\alpha_i(x_1) > \frac{m}{2}$ and $\alpha_j(x_1) < \frac{m}{2}$. Observe that $\Delta_p = 4$ and $\Delta_{Var} > 0$, implying that x_1' can be accepted by RLS.

Case (IV). $\alpha_i(x_1) \leq \frac{m}{2}$ and $\alpha_j(x_1) < \frac{m}{2}$. Observe that $\Delta_p = 0$, but the value of Δ_{Var} depends on the relationship between $\alpha_i(x_1)$ and $\alpha_j(x_1)$.

Summarizing the analysis of the four cases gets that during the optimization process, the potential value cannot increase. Observe that there exist $i, j \in [1, k]$ such that $\alpha_i(x_1) = |M_0(x_1) \cap G_i| > \frac{m}{2}$ and $\alpha_j(x_1) = |M_0(x_1) \cap G_j| < \frac{m}{2}$ (i.e., Case (III) holds), and the offspring obtained by the mutation flipping a 0-bit of G_i and a 1-bit of G_j in x_1 can be accepted. Now we consider the probability to generate such a mutation. Let $S_0 \subset [1, k]$ (resp., $S_1 \subset [1, k]$) such that for any $i \in S_0$, $\alpha_i(x_1) > \beta_i(x_1)$ (resp., $\alpha_i(x_1) < \beta_i(x_1)$). Since x_1 is an equal-solution,

$$\sum_{i \in S_0} \alpha_i(x_1) - \beta_i(x_1) = \sum_{i \in S_1} \beta_i(x_1) - \alpha_i(x_1) = p(x_1)/2. \tag{1}$$

Combining Equality (1) with $\sum_{i \in S_0} \alpha_i(x_1) + \beta_i(x_1) = |S_0|m$ and $\sum_{i \in S_1} \alpha_i(x_1) + \beta_i(x_1) = |S_1|m$ gets $\sum_{i \in S_0} \alpha_i(x_1) = \frac{p(x_1)}{4} + \frac{|S_0|m}{2} \geq \frac{p(x_1)}{4} + \frac{m}{2}$ and $\sum_{i \in S_1} \beta_i(x_1) = \frac{p(x_1)}{4} + \frac{|S_1|m}{2} \geq \frac{p(x_1)}{4} + \frac{m}{2}$. Thus there are $\frac{p(x_1)}{4} + \frac{m}{2}$ 0-bits, each of which is in a group G_u with $\alpha_u(x_1) > \frac{m}{2}$, and $\frac{p(x_1)}{4} + \frac{m}{2}$ 1-bits, each of which is in a group G_v with $\alpha_v(x_1) < \frac{m}{2}$. That is, RLS generates such a mutation with probability $\Omega((\frac{2m+p(x_1)}{4n})^2)$ and takes expected runtime $O((\frac{n}{2m+p(x_1)})^2)$ to obtain an offspring x_1' with $p(x_1') = p(x_1) - 4$. Considering all possible values for the potential of the maintained solution (note that $1 \leq p(x_1) \leq n$), the total expected runtime of RLS to obtain x^* based on x_1 can be upper bounded by

$$\sum_{t=1}^{n} O(\frac{n^2}{(t+2m)^2}) = O(n^2) \sum_{t=1}^{n} (t+2m)^{-2} = O(n^2) \int_1^n (t+2m)^{-2} dt = O(n^2/m).$$

In summary, RLS takes expected runtime $O(n^2/m) = O(kn)$ to obtain an optimal solution to I_1 based on the initial solution x_0. □

Theorem 3. *The expected runtime of (1+1) EA to obtain an optimal solution to the instance $I_1 = (a, c, d, \gamma, k, m)$ of CCMSP-1 is $O((k+m)n^2)$.*

Proof. As the mutation of (1+1) EA may flip more than two bits simultaneously, the reasoning given in Theorem 2 cannot be directly applied for the performance of (1+1) EA. We first consider the expected runtime of (1+1) EA to get the first equal-solution x_1 based on the initial solution x_0 that is assumed to have $|M_0(x_0)| > |M_1(x_0)|$. A vector function $v(x) = (|M_{t(x)}(x)|, b(x))$ is designed for the solutions x obtained during the process, where $b(x) = \sum_{i=1}^{k} \binom{|M_{t(x)}(x) \cap G_i|}{2}$.

For ease of notation, let $|M_{t(x)}(x)| = \ell$, where $\ell \in [\frac{n}{2}, n]$ (as $M_{t(x)}(x)$ is the fuller machine by Lemma 1). Then $0 < b(x) \leq \lfloor \frac{\ell}{m} \rfloor \binom{m}{2} + \binom{\ell \% m}{2} \leq (\frac{\ell}{m} + 1)\binom{m}{2}$, where the first $\leq$ holds by Observation 1. Hence the number of possible values of

$v(x)$ can be upper bounded by $\sum_{\ell=\frac{n}{2}+1}^{n}(\frac{\ell}{m}+1)\binom{m}{2} = O(mn^2)$. Observe that for any two solutions x and x', if $v(x) = v(x')$ then $L(x) = L(x')$. Thus the number of possible values of $L(x)$ can be upper bounded by $O(mn^2)$ as well.

Consider a mutation flipping a $t(x)$-bit on x (i.e., if $t(x) = 0$ then flipping a 0-bit; otherwise, a 1-bit). By the discussion for Case (1) given in Theorem 2, the solution x' obtained by the mutation has $L(x') < L(x)$ and can be accepted. The probability of (1+1) EA to generate such a mutation is $\Omega(1/2)$. Thus combining the probability and the number of possible values of $L(x)$ gives that (1+1) EA takes expected runtime $O(mn^2)$ to get the first equal-solution x_1 based on x_0.

Now we consider the runtime of (1+1) EA to obtain an optimal solution based on x_1. As all solutions accepted subsequently are equal-ones, we take $b(x)$ as the potential function, where the number of possible values of $b(x)$ can be bounded by $O(km^2)$. By the reasoning given in Theorem 2 a mutation flipping a 0-bit and a 1-bit obtaining an improved solution can be generated with probability $\Omega((\frac{m}{2n})^2)$. Consequently, (1+1) EA takes expected runtime $O(kn^2)$ to obtain an optimal solution based on x_1. In summary, (1+1) EA takes expected runtime $O((k+m)n^2)$ to obtain an optimal solution to I_1. □

5 Performance for CCMSP-2

The section starts with a lemma to show that the discussion for CCMSP-2 would be more complicated than that for CCMSP-1.

Lemma 3. *Given a solution x to an instance $I_2 = (a, c, d, \gamma, k, \{m_i | i \in [1, k]\})$ of CCMSP-2, whether $l'_0(x) > l'_1(x)$ holds is unknown even if $|M_0(x)| > |M_1(x)|$.*

Proof. Recall that the group G_i has size m_i, and there is a variable $\delta_i(x)$ such that $\alpha_i(x) = m_i/2 + \delta_i(x)$ and $\beta_i(x) = m_i/2 - \delta_i(x)$ for any $i \in [1, k]$. Thus

$$cov[l_0(x)] - cov[l_1(x)] = 2c\sum_{i=1}^{k}\left(\binom{\alpha_i}{2} - \binom{\beta_i}{2}\right) = 2c\textstyle\sum_{i=1}^{k}(m_i - 1)\delta_i.$$

Observe that $2c\sum_{i=1}^{k}(m_i-1)\delta_i$ can be treated as a weighted version of $\sum_{i=1}^{k}\delta_i(x)$, where $\sum_{i=1}^{k}\delta_i(x) > 0$ due to $|M_0(x)| > |M_1(x)|$, but it is impossible to decide whether $2c\sum_{i=1}^{k}(m_i-1)\delta_i$ is greater than 0. Furthermore, the relationship among the values of a, c and d are unrestricted. Consequently, it is also impossible to decide whether or not $l'_0(x) > l'_1(x)$ holds. □

For ease of analysis, we set an *extra constraint* on the values of a, c and d considered in the instances of CCMSP-2:

$$\sqrt{\frac{(1-\gamma)}{\gamma}\left(nd + 2c\sum_{i=1}^{k}\binom{m_i}{2}\right)} < a. \tag{2}$$

That is, for any solution x to any instance of CCMSP-2 and any $t \in [0, 1]$, $E[l_t(x)]$ contributes much more than $\sqrt{\frac{(1-\gamma)}{\gamma}Var[l_t(x)]}$ to $l'_t(x)$ under the extra constraint, because $\sqrt{\frac{(1-\gamma)}{\gamma} Var[l_t(x)]} \leq \sqrt{\frac{(1-\gamma)}{\gamma}\left(nd + 2c\sum_{i=1}^{k}\binom{m_i}{2}\right)} < a$. The new variant of CCMSP-2 is called CCMSP-2$^+$ in the remaining text.

Due to the extra constraint of CCMSP-2$^+$, for any solution x to I_2^+, if $|M_0(x)| > |M_1(x)|$ (resp., $|M_0(x)| < |M_1(x)|$) then $l'_0(x) > l'_1(x)$ (resp., $l'_0(x) < l'_1(x)$). Thus it is easy to derive the following lemma.

Lemma 4. *Given an instance $I_2^+ = (a, c, d, \gamma, k, \{m_i | i \in [1, k]\})$ of CCMSP-2$^+$, any optimal solution to I_2^+ is an equal-solution.*

Lemma 5. *CCMSP-2$^+$ is NP-hard.*

Proof. For the computational hardness of CCMSP-2$^+$, the discussion is divided based on the number of jobs considered in the instances of CCMSP-2$^+$.

Case 1. The instances of CCMSP-2$^+$ that consider odd many jobs.

Let $I_2^+ = (a, c, d, \gamma, k, \{m_i | i \in [1, k]\})$ be an instance of CCMSP-2$^+$, where $n = \sum_{i=1}^{k} m_i$ is odd. We construct an optimal solution x^* to I_2^+ as follows. By Lemma 4, x^* is an equal-solution. Assume $|M_0(x^*)| = |M_1(x^*)| + 1 = \frac{n+1}{2}$. By the extra constraint of CCMSP-2$^+$, $l'_0(x^*) > l'_1(x^*)$. Thus we only need to analyze the optimal allocation approach of $\frac{n+1}{2}$ many jobs on M_0 w.r.t. x^* such that $cov[l_0(x^*)]$ is minimized. By Observation 1, $k\binom{\frac{n+1}{2k}}{2} \leq cov[l_0(x^*)]$ (i.e., each group allocates $\frac{n+1}{2k}$ many jobs to M_0), but $\frac{n+1}{2k}$ may be not an integer. Fortunately, by Observation 1, it is easy to get that the optimal allocation approach of $\frac{n+1}{2}$ many jobs on M_0 w.r.t. x^* can be obtained as: For each $1 \leq i \leq k$ (assume that the values of $\alpha_j(x^*)$ for all $1 \leq j < i$ have been specified), if

$$m_i < (\frac{n+1}{2} - \sum_{j=1}^{i-1} \alpha_j(x^*))/(k+1-i),$$

then let $\alpha_i(x^*) = m_i$; otherwise, let $\alpha_i(x^*) = \lceil(\frac{n+1}{2} - \sum_{j=1}^{i-1} \alpha_j(x^*))/(k+1-i)\rceil$.

Observe that once $\alpha_i(x^*)$ is set as $\lceil(\frac{n+1}{2} - \sum_{j=1}^{i-1} \alpha_j(x^*))/(k+1-i)\rceil$, then for all $i < j \leq k$, $|\alpha_j(x^*) - \alpha_i(x^*)| \leq 1$ (as $m_1 \leq m_2 \leq \ldots \leq m_k$). In a word, the optimal solution x^* to I_2^+ satisfies the following property.

Property-Odd: For any $i \in [1, k]$, either $\alpha_i(x^*) = m_i$ or $0 \leq \alpha_{max}(x^*) - \alpha_i(x^*) \leq 1$, where $\alpha_{max}(x^*) = \max\{\alpha_1(x^*), \ldots, \alpha_k(x^*)\}$.

Case 2. The instances of CCMSP-2$^+$ that consider even many jobs.

It can be shown that any instance of the *Two-way Balanced Partition problem* can be polynomial-time reduced to an instance I_2^+ of CCMSP-2$^+$ such that I_2^+ has even many groups and each group has odd size, where the formulation of the Two-way Balanced Partition problem is: Given a multiset S that contains non-negative integers such that both $|S|$ and $\sum_{e \in S} e$ are even, can S be partitioned into two subsets S_1 and S_2 such that $|S_1| = |S_2|$ and $\sum_{a \in S_1} a = \sum_{b \in S_2} b$? The NP-hardness of the Two-way Balanced Partition problem can be shown by reducing it to the well-known Partition problem [6]. Due to the page limit, the detailed discussion will be given in a complete version. □

Corollary 1. *CCMSP-2 is NP-hard.*

5.1 Performance for CCMSP-2^+

Theorem 4. *Given an instance $I_2^+ = (a, c, d, \gamma, k, \{m_i | i \in [1, k]\})$ of CCMSP-2^+ that considers odd many jobs (i.e., $n = \sum_{i=1}^{k} m_i$ is odd), RLS takes expected runtime $O(\sqrt{k}n^3)$ to obtain an optimal solution to I_2^+.*

Proof. Let x_0 be the initial solution maintained by RLS. The optimization process of RLS for x_0 discussed below is divided into two phases.

Phase-1. Obtaining the first equal-solution x_1 based on x_0.

Let $p_1(x) = ||M_0(x)| - |M_1(x)||$ be the potential of the solution x maintained during Phase-1. Observe that $1 \leq p_1(x) \leq n$, and the extra constraint of CCMSP-2^+ indicates that for any two solutions x' and x'' to I_2^+, if $p_1(x') < p_1(x'')$ then $L(x') < L(x'')$. The mutation of RLS flipping exactly one bit in x whose corresponding job is allocated to the fuller machine w.r.t. x, can be generated by RLS with probability $\Omega(1/4)$, and the obtained solution x' has potential value $p_1(x') = p_1(x) - 2$. Combining $p_1(x') = p_1(x) - 2$ with the conclusion given above gets $L(x') < L(x)$, and x' can be accepted by RLS. Then using the Additive Drift analysis [8], we can derive that Phase-1 takes expected runtime $O(n)$. Note that after the acceptance of x_1, any non-equal-solution cannot be accepted. W.l.o.g., assume that $|M_0(x_1)| = |M_1(x_1)| + 1$.

Phase-2. Obtaining the first optimal solution based on x_1.

Case (1). $cov[l_0(x_1)] < cov[l_1(x_1)]$.

First of all, it is not hard to get that any mutation flipping exactly one bit of x_1 cannot get an improved solution under Case (1). Thus the following discussion only considers the mutations flipping a 0-bit of G_i and a 1-bit of G_j in x_1 (note that the other kinds of mutations flipping two bits cannot get equal-solutions). Denote by x_1' the obtained solution. Hence $|M_0(x_1')| = |M_0(x_1)| = |M_1(x_1')|+1 = |M_1(x_1)|+1$ and $cov[l_0(x_1)]-cov[l_0(x_1')] = 2c(\alpha_i(x_1)-1-\alpha_j(x_1))$. If $\alpha_i(x_1) - \alpha_j(x_1) \geq 1$ then $cov[l_0(x_1')] \leq cov[l_0(x_1)]$, and x_1' can be accepted.

Assume that RLS obtains a solution x_1^* based on x_1, on which all possible mutations flipping exactly a 0-bit and a 1-bit of x_1^* cannot get an improved solution, where the 0-bit and 1-bit are of G_i and G_j, respectively. Then x_1^* satisfies the property: For any $1 \leq i \neq j \leq k$, if $\alpha_i(x_1^*)-\alpha_j(x_1^*) \geq 2$, then all jobs of G_j are allocated to M_0 w.r.t. x_1^*, i.e., $\alpha_j(x_1^*) = m_j$ and $\beta_j(x_1^*) = 0$. In other words, for any $1 \leq j \leq k$, either $\alpha_j(x_1^*) = m_j$ or $0 \leq \alpha_{max}(x_1^*) - \alpha_j(x_1^*) \leq 1$, where $\alpha_{max}(x_1^*) = \max\{\alpha_1(x_1^*), \ldots, \alpha_k(x_1^*)\}$. Thus x_1^* satisfies Property-Odd given in the proof of Lemma 5, and x_1^* is an optimal solution to I_2^+.

For the expected runtime of RLS for Phase-2, let $p_{21}(x) = cov[l_0(x)]/2c$ be the potential of the solution x maintained during Phase-2. The above discussion shows that $|M_0(x)| = |M_1(x)| + 1$. Let $i_{max} = \arg\max\{\alpha_1(x), \ldots, \alpha_k(x)\}$. Then $\binom{\alpha_{i_{max}}(x)}{2} \geq \frac{p_{21}(x)}{k}$, implying that $\alpha_{i_{max}}(x) \geq (\sqrt{1 + \frac{8p_{21}(x)}{k}} + 1)/2$. Since x does not satisfy Property-Odd, there exists a $1 \leq j' \neq i_{max} \leq k$ such that $\alpha_{i_{max}}(x) - \alpha_{j'}(x) \geq 2$ but $\alpha_{j'}(x) < m_{j'}$. Thus $\beta_{j'}(x) \geq 1$. The mutation flipping a 0-bit of $G_{i_{max}}$ and a 1-bit of $G_{j'}$ in x can be generated by RLS with probability $\Omega(\frac{\alpha_{i_{max}}(x) \cdot \beta_{j'}(x)}{n^2}) = \Omega(\frac{\alpha_{i_{max}}(x)}{n^2}) = \Omega(\frac{1}{n^2}\sqrt{\frac{p_{21}(x)}{k}})$, and the potential value of the obtained solution is decreased by at least 1 compared to $p_{21}(x)$. Observe that the upper bound of $p_{21}(x_1)$ and lower bound of $p_{21}(x_1^*)$ are $\binom{\frac{n+1}{2}}{2}$ and $k\binom{\frac{n+1}{2k}}{2}$, respectively. Considering all possible potential values of x, we have that the expected runtime of RLS for Phase-2 can be bounded by

$$\sum_{t=k\binom{\frac{n+1}{2k}}{2}}^{\binom{\frac{n+1}{2}}{2}} O(\frac{\sqrt{k}n^2}{\sqrt{t}}) = O(\sqrt{k}n^2)\int_{k\binom{\frac{n+1}{2k}}{2}}^{\binom{\frac{n+1}{2}}{2}} t^{-\frac{1}{2}}dt = O(\sqrt{k}n^3).$$

Case (2). $cov[l_0(x_1)] \geq cov[l_1(x_1)]$.

The main difference between the discussion for Case (2) and that for Case (1) is that the mutation flipping one bit may generate an improved solution, implying that the fuller machine may be M_0 or M_1. However, no matter which one is the fuller machine, the value $cov[l_{t(x)}(x)]$ cannot increase during Phase-2, where x is a solution maintained by RLS during Phase-2. By the reasoning given for Case (1), for a mutation flipping exactly one 0-bit of G_i and one 1-bit of G_j in x, if $t(x) = 0$ and $\alpha_i(x) - \alpha_j(x) \geq 2$, or $t(x) = 1$ and $\beta_j(x) - \beta_i(x) \geq 2$, then $cov[l_{t(x')}(x')] < cov[l_{t(x)}(x)]$ for the obtained solution x', and x' can be accepted.

Let $p_{22}(x) = cov[l_{t(x)}(x)]/2c$ be the potential of the solution x. Using the reasoning similar to that given for Case (1), we can get that RLS takes expected runtime $O(\sqrt{k}n^3)$ to obtain an optimal solution to I_2^+ under Case (2). $\square$

Theorem 5. *Given an instance $I_2^+ = (a, c, d, \gamma, k, \{m_i | i \in [1, k]\})$ of CCMSP-2^+ that considers even many jobs (i.e., $n = \sum_{i=1}^k m_i$ is even), RLS takes expected runtime $O(n^4)$ to obtain an equal-solution x^* such that either $|cov[l_0(x^*)] - cov[l_1(x^*)]| \leq 2c(m_k - m_1 - 1)$ or $cov[l_{t(x^*)}(x^*)] \leq \frac{c}{4}(\frac{n^2}{k} - 2n + k)$.*

Proof. Let x_0 be the initial solution maintained by RLS. The proof runs in a similar way to that of Theorem 4, dividing the optimization process into two phases: **Phase-1**, obtaining the first equal-solution x_1 based on x_0; **Phase-2**, optimizing the solution x_1. Moreover, the analysis for Phase-1 is the same as that given in the proof of Theorem 4, i.e., Phase-1 takes expected runtime $O(n)$. Now we consider Phase-2, where the solution x_1 is assumed to have $cov[l_0(x_1)] > cov[l_1(x_1)]$. Let $\Delta(x_1) = cov[l_0(x_1)] - cov[l_1(x_1)]$. The following discussion only considers the mutations flipping a 0-bit of G_i and a 1-bit of G_j in x_1. Denote by x_1' the obtained solution. We have

$$\begin{aligned}&cov[l_0(x_1')]\\ =&cov[l_0(x_1)] - 2c\left[\left(\binom{\alpha_i(x_1)}{2} + \binom{\alpha_j(x_1)}{2}\right) - \left(\binom{\alpha_i(x_1)-1}{2} + \binom{\alpha_j(x_1)+1}{2}\right)\right]\\ =&cov[l_0(x_1)] + 2c(\alpha_j(x_1) - \alpha_i(x_1) + 1)\end{aligned}$$

and $cov[l_1(x_1')] = cov[l_1(x_1)] + 2c(\beta_i(x_1) - \beta_j(x_1) + 1)$ similarly.

If $\alpha_j(x_1) \leq \alpha_i(x_1) - 1$ (i.e., $cov[l_0(x_1')] \leq cov[l_0(x_1)]$) and $\beta_i(x_1) - \beta_j(x_1) + 1 \leq \Delta(x_1)/2c$ (i.e., $cov[l_1(x_1')] \leq cov[l_0(x_1)]$), then $L(x_1') \leq L(x_1)$ and x_1' can be accepted by RLS, and $cov[l_0(x_1')] - cov[l_1(x_1')] = \Delta(x_1) + 2c(m_j - m_i)$.

Now we assume that RLS obtains a solution x_1^* based on x_1 such that any mutation flipping a 0-bit and a 1-bit of x_1^* cannot get an improved solution, and $cov[l_0(x_1^*)] \geq cov[l_1(x_1^*)]$. Let $i_{max} = \arg\max\{\alpha_1(x_1^*), \alpha_2(x_1^*), \ldots, \alpha_k(x_1^*)\}$. Then the above discussion shows that for any $j \in [1,k]$, if $\alpha_j(x_1^*) < \alpha_{i_{max}}(x_1^*) - 1$ then $\beta_{i_{max}}(x_1^*) - \beta_j(x_1^*) + 1 \geq \Delta(x_1^*)/2c$, i.e.,

$$(m_{i_{max}} - \alpha_{i_{max}}(x_1^*)) - (m_j - \alpha_j(x_1^*)) \geq \Delta(x_1^*)/2c - 1,$$

implying that (recall that $m_1 \leq m_2 \leq \ldots \leq m_k$)

$$\Delta(x_1^*)/2c + 1 \leq \Delta(x_1^*)/2c - 1 + (\alpha_{i_{max}}(x_1^*) - \alpha_j(x_1^*)) \leq m_{i_{max}} - m_j \leq m_k - m_1.$$

In other words, for x_1^*, if there is a $j \in [1,k]$ with $\alpha_j(x_1^*) < \alpha_{i_{max}}(x_1^*) - 1$, then $\Delta(x_1^*)/2c \leq m_k - m_1 - 1$. If there is no $j \in [1,k]$ with $\alpha_j(x_1^*) < \alpha_{i_{max}}(x_1^*) - 1$, then for each $j \in [1,k]$, $0 \leq \alpha_{i_{max}}(x_1^*) - \alpha_j(x_1^*) \leq 1$. Now we bound the value of $cov[l_0(x_1^*)]$. Let $\tau = |\{1 \leq j \leq k | \alpha_j(x_1^*) = \alpha_{i_{max}}(x_1^*) - 1\}|$. Then $(k-\tau)\alpha_{i_{max}}(x_1^*) + \tau(\alpha_{i_{max}}(x_1^*) - 1) = n/2$ implies that $\alpha_{i_{max}}(x_1^*) = \frac{n}{2k} + \frac{\tau}{k}$, and

$$\begin{aligned} cov[l_0(x_1^*)]/2c &= (k-\tau)\binom{\alpha_{i_{max}}(x_1^*)}{2} + \tau\binom{\alpha_{i_{max}}(x_1^*) - 1}{2} \\ &= \frac{n^2}{8k} - \frac{n}{4} - (\frac{\tau^2}{2k} - \frac{\tau}{2}) \leq \frac{n^2}{8k} - \frac{n}{4} + \frac{k}{8}, \end{aligned}$$

where $\frac{\tau^2}{2k} - \frac{\tau}{2}$ gets its minimum value $-\frac{k}{8}$ when $\tau = \frac{k}{2}$.

For the expected runtime of RLS to get x_1^* based on x_1, let $p(x) = cov[l_{t(x)}(x)]/2c$ be the potential of x that is a solution maintained by RLS during the process. Observe that $p(x)$ cannot increase during the process. The probability of RLS to generate such a mutation mentioned above is $\Omega(1/n^2)$, and the potential value decreases by at least 1. As $p(x_1)$ can be upper bounded by $O(n^2)$, using the Additive Drift analysis [8] gets that RLS takes expected runtime $O(n^4)$ to obtain x_1^* based on x_1. In summary, RLS takes expected runtime $O(n^4)$ to obtain an equal-solution x_1^* satisfying the claimed condition based on x_0. □

6 Conclusion

The paper studied a chance-constrained version of the Makespan Scheduling problem and investigated the performance of RLS and (1+1) EA for it. More specifically, the paper studied two simple variants of the problem (namely, CCMSP-1 and CCMSP-2$^+$) and obtained a series of results: CCMSP-1 was shown to be polynomial-time solvable by giving the expected runtime of RLS and (1+1) EA to obtain an optimal solution to the given instance of CCMP-1; CCMSP-2$^+$ was shown to be NP-hard by reducing the Two-way Balanced Partition problem to it, but any instance of CCMSP-2$^+$ which considers odd many jobs was shown to be polynomial-time solvable by giving the expected runtime of RLS to obtain an optimal solution to it.

Future work on the Chance-constrained Makespan Scheduling problem or the chance-constrained version of other classical combinatorial optimization problems would be interesting, and these related results would further advance and broaden the understanding of evolutionary algorithms.

References

1. Blazewicz, J., Lenstra, J., Kan, A.: Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics **5**(1), 11–24 (1983)
2. Charnes, A., Cooper, W.W.: Chance-constrained programming. Management science **6**(1), 73–79 (1959)
3. Corus, D., Lehre, P.K., Neumann, F.: The generalized minimum spanning tree problem: A parameterized complexity analysis of bi-level optimisation. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 519–526. ACM (2013)
4. Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Analyses of simple hybrid algorithms for the vertex cover problem. Evolutionary Computation **17**(1), 3–19 (2009)
5. Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Approximating covering problems by randomized search heuristics using multi-objective models. Evolutionary Computation **18**(4), 617–633 (2010)
6. Hayes, B.: Computing science: The easiest hard problem. American Scientist **90**(2), 113–117 (2002)
7. He, J., Mitavskiy, B., Zhou, Y.: A theoretical assessment of solution quality in evolutionary algorithms for the knapsack problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC). pp. 141–148. IEEE (2014)
8. He, J., Yao, X.: A study of drift analysis for estimating computation time of evolutionary algorithms. Natural Computing **3**, 21–35 (2004)
9. Iwamura, K., Liu, B.: A genetic algorithm for chance constrained programming. Journal of Information and Optimization Sciences **17**(2), 409–422 (1996)
10. Jansen, T., Oliveto, P.S., Zarges, C.: Approximating vertex cover using edge-based representations. In: Proceedings of the Workshop on Foundations of Genetic Algorithms. pp. 87–96. ACM (2013)
11. Kratsch, S., Lehre, P.K., Neumann, F., Oliveto, P.S.: Fixed parameter evolutionary algorithms and maximum leaf spanning trees: A matter of mutation. In: Proceedings of the Conference on Parallel Problem Solving from Nature. pp. 204–213. Springer (2010)
12. Kratsch, S., Neumann, F.: Fixed-parameter evolutionary algorithms and the vertex cover problem. Algorithmica **65**(4), 754–771 (2013)
13. Lissovoi, A., Witt, C.: A runtime analysis of parallel evolutionary algorithms in dynamic optimization. Algorithmica **78**(2), 641–659 (2017)
14. Miller, B.L., Wagner, H.M.: Chance constrained programming with joint constraints. Operations Research **13**(6), 930–945 (1965)
15. Neumann, A., Neumann, F.: Optimising monotone chance-constrained submodular functions using evolutionary multi-objective algorithms. In: Proceedings of the 16th International Conference on Parallel Problem Solving from Nature (PPSN). pp. 404–417. Springer (2020)
16. Neumann, F., Pourhassan, M., Roostapour, V.: Analysis of evolutionary algorithms in dynamic and stochastic environments, pp. 323–357. Springer International Publishing, Cham (2020)
17. Neumann, F., Sutton, A.M.: Runtime analysis of the (1+1) evolutionary algorithm for the chance-constrained knapsack problem. In: Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA). pp. 147–153 (2019)

18. Neumann, F., Wegener, I.: Minimum spanning trees made easier via multi-objective optimization. Natural Computing **5**, 305–319 (2006)
19. Neumann, F., Wegener, I.: Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. Theoretical Computer Science **378**(1), 32–40 (2007)
20. Neumann, F., Witt, C.: On the runtime of randomized local search and simple evolutionary algorithms for dynamic makespan scheduling. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 3742–3748 (2015)
21. Oliveto, P.S., He, J., Yao, X.: Analysis of population-based evolutionary algorithms for the vertex cover problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 1563–1570. IEEE (2008)
22. Oliveto, P.S., He, J., Yao, X.: Analysis of the (1+1) EA for finding approximate solutions to vertex cover problems. IEEE Transactions on Evolutionary Computation **13**(5), 1006–1029 (2009)
23. Poojari, C.A., Varghese, B.: Genetic algorithm based technique for solving chance constrained problems. European journal of operational research **185**(3), 1128–1154 (2008)
24. Pourhassan, M., Friedrich, T., Neumann, F.: On the use of the dual formulation for minimum weighted vertex cover in evolutionary algorithms. In: Proceedings of the Workshop on Foundations of Genetic Algorithms. pp. 37–44. ACM (2017)
25. Pourhassan, M., Gao, W., Neumann, F.: Maintaining 2-approximations for the dynamic vertex cover problem using evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 513–518 (2015)
26. Pourhassan, M., Shi, F., Neumann, F.: Parameterized analysis of multi-objective evolutionary algorithms and the weighted vertex cover problem. In: Proceedings of the Conference on Parallel Problem Solving from Nature. pp. 729–739. Springer (2016)
27. Roostapour, V., Neumann, A., Neumann, F., Friedrich, T.: Pareto optimization for subset selection with dynamic cost constraints. Artificial Intelligence **302**, 103597 (2022)
28. Shi, F., Neumann, F., Wang, J.: Runtime performances of randomized search heuristics for the dynamic weighted vertex cover problem. Algorithmica **83**(4), 906–939 (2021)
29. Shi, F., Schirneck, M., Friedrich, T., Kötzing, T., Neumann, F.: Reoptimization time analysis of evolutionary algorithms on linear functions under dynamic uniform constraints. Algorithmica **81**(2), 828–857 (2019)
30. Sutton, A.M., Neumann, F.: A parameterized runtime analysis of simple evolutionary algorithms for makespan scheduling. In: Proceedings of the Conference on Parallel Problem Solving from Nature. pp. 52–61. Springer (2012)
31. Vahid, R., Jakob, B., Neumann, F.: Runtime analysis of evolutionary algorithms with biased mutation for the multi-objective minimum spanning tree problem. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 551–559. ACM (2020)
32. Witt, C.: Worst-case and average-case approximations by simple randomized search heuristics. In: Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science. pp. 44–56. Springer (2005)
33. Witt, C.: Revised analysis of the (1+1) EA for the minimum spanning tree problem. In: Proceedings of the Conference on Genetic and Evolutionary Computation. pp. 509–516. ACM (2014)

34. Xie, Y., Harper, O., Assimi, H., Neumann, A., Neumann, F.: Evolutionary algorithms for the chance-constrained knapsack problem. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). pp. 338–346 (2019)
35. Xie, Y., Neumann, A., Neumann, F., Sutton, A.M.: Runtime analysis of RLS and the (1+1) EA for the chance-constrained knapsack problem with correlated uniform weights. arXiv preprint arXiv:2102.05778 (2021)
36. Yu, Y., Yao, X., Zhou, Z.H.: On the approximation ability of evolutionary optimization with application to minimum set cover. Artificial Intelligence **180**, 20–33 (2012)