ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230594

Optimizing Chance-Constrained Submodular Problems with Variable Uncertainties

Xiankun Yan[a;*], Anh Viet Do[a], Feng Shi[b,c], Xiaoyu Qin[d] and Frank Neumann[a]

[a]Optimisation and Logistics, University of Adelaide, Adelaide, Australia
[b]School of Computer Science and Engineering, Central South University, Changsha, P.R. China
[c]Xiangjiang Laboratory, Changsha, P.R. China
[d]University of Birmingham, Birmingham, UK

Abstract. Chance constraints are frequently used to limit the probability of constraint violations in real-world optimization problems where the constraints involve stochastic components. We study chance-constrained submodular optimization problems, which capture a wide range of optimization problems with stochastic constraints. Previous studies considered submodular problems with stochastic knapsack constraints in the case where uncertainties are the same for each item that can be selected. However, uncertainty levels are usually variable with respect to the different stochastic components in real-world scenarios, and rigorous analysis for this setting is missing in the context of submodular optimization. This paper provides the first such analysis for this case, where the weights of items have the same expectation but different dispersion. We present greedy algorithms that can obtain a high-quality solution, i.e., a constant approximation ratio to the given optimal solution from the deterministic setting. In the experiments, we demonstrate that the algorithms perform effectively on several chance-constrained instances of the maximum coverage problem and the influence maximization problem.

1 Introduction

Stochastic components can significantly affect the quality of solutions for a given stochastic optimization problem. Reducing the uncertain effect of stochastic components is vital to avoid potentially disruptive incidents in the complex and expensive system. *Chance constraints* can be applied to optimization tasks, which limit the probability of incidental constraint violations [1, 7, 13, 18]. A chance-constrained optimization problem can be described as finding an optimal solution subject to the condition that the constraints are only violated with a given small probability. Recently, the problem has been investigated widely [4, 15, 16, 17, 23, 24, 25, 26]. A typical technique for taking chance constraints into account for a given optimization problem is to convert the stochastic constraints to their respective deterministic equivalents for a given confidence level, which is possible when considering normally distributed stochastic components.

Submodular functions [14] capture problems of diminishing returns which frequently appear in real-world scenarios. They constitute a significant category of optimization challenges. In the artificial intelligence literature, greedy algorithms[3, 4, 6, 28] and Pareto optimization approaches[15, 21, 19, 20] based on evolutionary multi-objective algorithms have been widely examined for submodular optimization problems. The goal for a submodular optimization problem with a given knapsack constraint is to find a set of elements with the maximal value of the submodular function whose total weight does not exceed the budget of the given knapsack. There are many analyses on the deterministic version of this submodular optimization problem [9, 10, 14, 19]. Often the weights of elements might be stochastic and sampled from a probability distribution. The *Chance-constrained Submodular Problem* [2] has been proposed to model this case. Here, the goal of the problem is to maximize a given monotone submodular function subject to the constraint that the probability of violating the knapsack constraint is no more than a small threshold value. For this problem, Chen and Maehara [2] reduced the chance constraint of the problem into multiple deterministic constraints by guessing the parameters and relaxed the knapsack budget and threshold. They rigorously analyzed an algorithm that enumerates all parameters for the abstracted problem with random weights sampled from arbitrary known distributions, which meets the relaxed constraint. Doerr et al., [4] investigated a specific variant of the problem where the weights are sampled from a uniform distribution with an identical dispersion. They applied the one-sided Chebyshev's inequality and a Chernoff bound separately to construct the surrogates that helps to estimate the probability of constraint violation. In addition, they empirically showed that using the greedy algorithms based on such surrogates gives high-quality solutions in stochastic scenarios. Furthermore, multi-objective evolutionary algorithms have also been employed to tackle this problem, e.g., the GSEMO algorithm [15]. It has been theoretically analyzed and found to achieve comparable performance to the greedy algorithms in the worse case within polynomial time. However, uncertainties of items usually vary between the items and greedy algorithms with theoretical performance guarantees missing in the literature. Such an analysis is supposed to be more challenging than the one carried out in [4] since variable uncertainties of the weights lead to more intricate effects than identical uncertainties, which is reflected in the surrogate based on one-sided Chebyshev's inequality.

In this paper, we focus on a general setting of the problem studied in [4, 15], i.e., the weights of the elements are sampled from uniform distributions with the same expectation but different dispersion values, instead of from an identical uniform distribution. We

[*] Corresponding Author. Email: xiankun.yan@adelaide.edu.au.

remark the item's dispersion as its uncertainty level, such that the uncertainty level varies from item to item in this setting. The one-sided Chebyshev's inequality is used to construct a surrogate of the chance constraint. In addition to the greedy algorithm (GA) and the generalized greedy algorithm (GGA), our analysis also encompasses another studied greedy algorithm, the generalized greedy+Max algorithm (GGMA) [27]. Our rigorous analysis demonstrates that the GA struggles to effectively obtain an acceptable solution in the worse case due to a heavy impact arising from the variable uncertainties. For the GGA and the GGMA, we first use a simple strategy for element selection, which only considers the sum of the dispersion values. The algorithms cannot guarantee a high-quality solution in some linear instances. Instead of using this simple strategy, simultaneously considering the expectation and the dispersion is promising to fill this gap. We adopt an improved strategy that applies the surrogate of the chance constraints in selecting elements. Using this strategy, the GGA and the GGMA can obtain a $(1/2-o(1))(1-1/e)-$approximation and $(1/2-o(1))-$approximation, respectively. Finally, we empirically analyze the performance of the algorithms on twelve chance-constrained instances of the maximum coverage problem and the influence maximization problem. The empirical results show that the GGA and the GGMA beat the GA in most instances. Furthermore, the GGMA obtains a solution of similar quality as the GGA, which verifies and supplements our theoretical results.

The paper is structured as follows. Sections 2-3 introduce the studied problem and the algorithms. Our theoretical results of the investigated algorithms are shown in Sections 4-6. We present our experimental results in Section 7 and finish with some conclusions in Section 8.

2 Preliminaries

2.1 Problem Definition

Consider a set $V=\{1,...,n\}$, in which each element $i\in V$ has a weight w_i, and a function $f: V'\to\mathbb{R}_{\geq 0}$ defined on the subsets $V'\subseteq V$. The function f is *monotone* iff for any two subsets $S,T\subseteq V$ with $S\subseteq T$, $f(S)\leq f(T)$ holds. Besides, the function f is *submodular* iff for any two subsets $S,T\subseteq V$ with $S\subseteq T$ and any element $e\notin T$,

$$f(S\cup\{e\})-f(S)\geq f(T\cup\{e\})-f(T). \tag{1}$$

Given a *monotone* submodular function f defined on the subsets of V and a budget B, the problem named *the submodular problem with respect to* V *and* B is to look for a subset $S\subseteq V$ such that $f(S)$ is maximized and $W(S)\leq B$, where $W(S)=\sum_{i\in S}w_i$.

Within the investigation, we focus on a *chance-constrained* version of the submodular problem, in which the weight w_i of each element $i\in V$ is random (not deterministic) and has expected value $E[w_i]=a_i$ and variance $\sigma_i^2\geq 0$. The aim is find a subset $S\subseteq V$ such that $f(S)$ is maximized and subject to the constraint that $Pr[W(S)>B]\leq\alpha$, where the threshold $0\leq\alpha\leq 1$ is given which upper bounds the probability of a constraint violation.

As mentioned above, given an instance of the chance-constrained submodular problem, the weight $W(S)=\sum_{i\in S}w_i$ of a solution S to it is random but has an expectation $E[W(S)]=\sum_{i\in S}a_i$, and variance $Var[W(S)]=\sum_{i\in S}\sigma_i^2$.

Within the paper, we consider a specific setting for the chance-constrained submodular problem, in which the random weight w_i of the element $i\in V$ is independently uniformly sampled from the interval $[a_i-\delta_i, a_i+\delta_i]$ at random. Besides, we consider $a_i=1$ and $0\leq\delta_i\leq 1$. Note that, for a uniform distribution, the expectation and variance can be calculated by the given interval bounds [24]. Therefore the expected weights of all elements are 1, and the variance of each element i is $Var[W(i)]=\delta_i^2/3$. Furthermore, w.l.o.g., assume every single element is feasible with respect to the budget B. Observe that the case that $0\leq B\leq 1$ is meaningless, thus we assume $B>1$ such that at least one item is in the solution throughout the paper.

2.2 Surrogate of the Chance constraint

For the probability $Pr[W(S)>B]$, as the work given in [24], we consider the one-sided Chebyshev's inequality to construct a usable surrogate of the chance constraint, whose formulation is given below.

Theorem 1 (*One-sided Chebyshev's inequality*)**.** *For any random variable* X *and* $\lambda\geq 0$, $Pr[X>E[X]+\lambda]\leq\frac{Var[X]}{Var[X]+\lambda^2}$.

For a solution S to the chance-constrained submodular problem, if $Pr[W(S)>B]\leq\alpha$, then it is *feasible*; otherwise, *infeasible*. By the one-sided Chebyshev's inequality, we have the following observation directly, which considers the feasibility of a given solution.

Observation 1. *Given a solution* S *to the chance-constrained submodular problem, if* $E[W(S)]+\sqrt{\frac{(1-\alpha)Var[W(S)]}{\alpha}}\leq B$ *then the solution* S *is feasible.*

By the above observation, the *surrogate weight* of a solution can be defined as $\Gamma(S):=E[W(S)]+\kappa_\alpha\sqrt{Var[W(S)]}$, where $\kappa_\alpha=\sqrt{\frac{1-\alpha}{\alpha}}$.

3 Algorithms

The first algorithm studied is the greedy algorithm (GA, see Algorithm 1), which was analyzed in [4] for the chance-constrained submodular problem with all elements having iid weights. The GA starts with an empty set and picks the element with the largest marginal gain that meets the constraint in each iteration. It stops when no more elements can be accepted without violating the constraint.

Considering that the elements may have different weights, the generalized greedy algorithm (GGA, see Algorithm 2) is studied. Similar to the mechanism of the GA, the GGA starts with an empty set and stops when no more elements can be added due to the chance constraint. However, the GGA selects the element that satisfies the chance constraint and maximizes the ratio between the additional gain in the objective function f and that in a non-decreasing function h. As the variances and surrogate weights of the solutions to the problem are non-decreasing, two strategies based on two non-decreasing functions h are respectively studied: **Strategy I** $h(S):=\sum_{i\in S}\delta_i^2$, and **Strategy II** $h(S):=\Gamma(S)$. Uncertainties of the solution are only considered in Strategy I and the surrogate weight based on the one-sided Chebyshev's inequality is studied in Strategy II. Furthermore, Lines 9-10 of Algorithm 2 are required when there exists an element with an extremely high objective value, see [9, 11] for more details.

Additionally, the generalized greedy+Max algorithm (GGMA, see Algorithm 3) is studied. The GGMA adopts the same greedy strategies as the GGA, but it uses the feasible item having the largest marginal gain to augment every partial greedy solution. More specifically, the augmenting item is selected among the remaining items that still meet the constraint in each iteration. Until no element can

Algorithm 1 Greedy Algorithm (GA)

```
Input: Elements set $V$, budget constraint $B$, failure probability $\alpha$
Output: $S$
1: $S \leftarrow \emptyset, V' \leftarrow V$
2: while $V' \neq \emptyset$ do
3:     $v^* \leftarrow \arg\max_{v \in V'} f(S \cup \{v\}) - f(S)$
4:     if $\Gamma(S \cup \{v^*\}) \leq B$ then
5:         $S \leftarrow S \cup \{v^*\}$
6:     end if
7:     $V' \leftarrow V' \setminus \{v^*\}$
8: end while
```

Algorithm 2 Generalized Greedy Algorithm (GGA)

```
Input: Elements set $V$, budget constraint $B$, failure probability $\alpha$
Output: $S$
1: $S \leftarrow \emptyset, V' \leftarrow V$
2: while $V' \neq \emptyset$ do
3:     $v^* \leftarrow \arg\max_{v \in V'} \frac{f(S \cup \{v\}) - f(S)}{h(S \cup \{v\}) - h(S)}$
4:     if $\Gamma(S \cup \{v^*\}) \leq B$ then
5:         $S \leftarrow S \cup \{v^*\}$
6:     end if
7:     $V' \leftarrow V' \setminus \{v^*\}$
8: end while
9: $v^* \leftarrow \arg\max_{\{v \in V; Pr[W(v) > B] \leq \alpha\}} f(v)$
10: $S \leftarrow \arg\max_{Y \in \{S, \{v^*\}\}} f(Y)$
```

fit into the solution, the GGMA stops and outputs the best-augmented solution.

Note that the surrogate is applied to the algorithms instead of calculating the probability $Pr[W(S) > B]$. We only are using the exact calculation for $Pr[W(v) > B]$ when considering a single element at line 9 in the GGA.

4 Performance of the GA

According to the previous work [4], the GA is theoretically proven that works well in chance-constrained submodular problems with identical weight and uncertainty. However, since the uncertainties become variable, the GA is hard to obtain a high-quality solution, which is proved in the below.

Let S_{cc} be the solution obtained by the GA. From Theorem 2, we find that the GA performs badly on some linear instances. Before the statement, we define such a linear instance I_1, in which let V have at least $B+1$ elements, $f(S) = |S|$, $\gamma \in (0,1]$, $\alpha \in \left(\frac{3\gamma}{(B-1)^2+3\gamma}, \frac{3\gamma}{(B-2)^2+3\gamma}\right)$, $\delta_1 = \sqrt{\gamma}$, and $\delta_i = 0$ for all $i \geq 2$.

Theorem 2. *There exists a linear instance I_1 such that the GA fails to guarantee better than $(1/B)$-approximation.*

Proof. Considering the instance I_1, we have $\Gamma(\{1\}) \in (B-1, B)$ and the GA on I_1 can pick element 1 in the first iteration, preventing it from continuing. Thus $f(S_{cc}) = 1$, and the solution is $(1/B)-$approximation while $Y = \{2, \ldots, B+1\}$, $f(Y) = \Gamma(Y) = B$. The claim is proved. □

The proof reveals that the GA rapidly exhausts the budget (i.e., selecting only one element) due to the significant influence of dispersion in the surrogate. This is the primary factor leading to the suboptimal performance of the GA.

Algorithm 3 Generalized Greedy+Max Algorithm (GGMA)

```
Input: Elements set $V$, budget constraint $B$, failure probability $\alpha$
Output: $T$
1: $T \leftarrow \emptyset, S \leftarrow \emptyset, V' \leftarrow V$
2: $V' \leftarrow \{v \in V' \setminus S \mid \Gamma(S \cup \{v\}) \leq B\}$
3: while $V' \neq \emptyset$ do
4:     $v' \leftarrow \arg\max_{v \in V'} f(S \cup \{v\})$
5:     if $f(T) < f(S \cup \{v'\})$ then
6:         $T \leftarrow S \cup \{v'\}$
7:     end if
8:     $v^* \leftarrow \arg\max_{v \in V'} \frac{f(S \cup \{v\}) - f(S)}{h(S \cup \{v\}) - h(S)}$
9:     $S \leftarrow S \cup \{v^*\}$
10:    Update $V'$ as Line 2
11: end while
```

5 Performance of the GGA

5.1 Analysis of Using Strategy I

For Strategy I: $h(S) := \sum_{i \in S} \delta_i^2$, we also find that there exists a collection of linear instances of the problem, for which the GGA is hard to obtain a high-quality solution. To facilitate the construction of these instances, a solution S is encoded as a decision vector $X = x_1 x_2 \ldots x_n$ with length n, where $x_i = 1$ means that the element $i \in V$ is selected into the solution S. Then we define such an instance I_2 with a linear function f, in which let $V = 1, \ldots, n$, $0 < \alpha < 0.5$, and $B = \varepsilon + 1$ where $n \geq 2\varepsilon$ and $\varepsilon \geq 1$. The function f represented by the decision vector X is given as:

$$f(X) = \sum_{i=1}^{\varepsilon} x_i + \varepsilon \sum_{i=\varepsilon+1}^{n} x_i. \tag{2}$$

Besides the dispersion of each element in I_2 is considered as $\delta_i = \sqrt{\frac{\gamma}{\varepsilon}}$ for $i \in [1, \varepsilon]$, and $\delta_j = \sqrt{\frac{\varepsilon\gamma + \beta}{\varepsilon}}$ for $j \in [\varepsilon + 1, n]$ subjected to $0 < \gamma$, $0 < \beta$ and $\varepsilon\gamma + \beta \leq 3\alpha/(1-\alpha)$, which indicated by Theorem 3 (proof in Appendix **??**).

Theorem 3. *Given $\varepsilon \geq 1$, there exists a linear instance I_2 such that the generalized greedy algorithm GGA applying $h := \sum_{i \in S} \delta_i^2$ fails to guarantee better than $(1/\varepsilon)-$approximation.*

5.2 Analysis of Using Strategy II

Since $h(S) := \Gamma(S)$ in Strategy II, we know that h is a non-linear function therefore the surrogate weight of each element is changed as the size of the solution grows. For the analysis (Theoreom 4), some useful notations and definitions are introduced first. Let S_{cc} be the greedy solution generated by the GGA, v_i be the i-th element added to the solution S_{cc}, and $S_i = \{v_1, \ldots v_i\} \subseteq S_{cc}$ $(1 \leq i \leq |S_{cc}|)$ be the set containing the first i elements. Then we define a set A_i to collect all abandoned elements due to the constraint violation before the GGA adds v_i into S_{i-1}. Note that $A_{i-1} \subseteq A_i$. Besides, the surrogate weight of the element v_i is denoted by c_i, where $c_i = \Gamma(S_i) - \Gamma(S_{i-1})$. Moreover, given any two sets $S, T \subset V$, let $f(S \mid T) := f(S \cup T) - f(T)$.

Let OPT_d be the optimal solution of the deterministic instance of the problem. Given a partial greedy solution S_k generated by the GGA, the relation between S_k and OPT_d is first investigated in Lemma 1 (proof in Appendix **??**). Observe that $|OPT_d| = \lfloor B \rfloor$ as the expected weight is exactly one.

Lemma 1. *Let $\zeta = \kappa_\alpha \sum_{j \in OPT_d} \sqrt{\delta_j^2/3}$. Given a partial greedy solution S_k, if $A_k \cap OPT_d = \emptyset$, then*

$$f(S_{k+1}) - f(S_k) \geq \frac{c_{k+1}}{\lfloor B \rfloor + \zeta} \cdot (f(OPT_d) - f(S_k)).$$

After that, we can get a relation between OPT_d and S_{cc} by using Lemma 1.

Theorem 4. *The solution obtained by the GGA applying $h := \Gamma(S)$ is a $(1/2 - o(1))(1 - 1/e)-$approximation.*

Proof. Consider the upper bound of k that is denoted by k^*. It has the set S_{k^*} such that the element from OPT_d is first abandoned due to the constraint when the GGA attempts to add it into the set. We denote the abandoned element by z and derive a relation between S_{k^*} and OPT_d.

Note that $A_i \cap OPT_d = \emptyset$ for $1 \leq i \leq k^*$. Following Lemma 1, it gives

$$f(S_{k^*+1}) - f(S_{k^*}) \geq \frac{c_{k+1}}{\lfloor B \rfloor + \zeta} \cdot (f(OPT_d) - f(S_{k^*})). \tag{3}$$

As we know that $(1 - x) \leq e^{-x}$, then rearranging (3) gives

$$\begin{aligned} & f(OPT_d) - f(S_{k^*+1}) \\ & \quad \leq \left(1 - \frac{c_{k^*+1}}{\lfloor B \rfloor + \zeta}\right) \cdot (f(OPT_d) - f(S_{k^*})) \\ & \quad \leq e^{-\frac{c_{k^*+1}}{\lfloor B \rfloor + \zeta}} \cdot (f(OPT_d) - f(S_{k^*})). \end{aligned} \tag{4}$$

Recursively,

$$\begin{aligned} & f(OPT_d) - f(S_{k^*+1}) \\ & \quad \leq e^{-\frac{c_{k^*+1}}{\lfloor B \rfloor + \zeta}} \cdot (f(OPT_d) - f(S_{k^*})) \\ & \quad \leq e^{-\frac{c_{k^*+1} + c_{k^*}}{\lfloor B \rfloor + \zeta}} \cdot (f(OPT_d) - f(S_{k^*-1})) \\ & \quad \leq \ldots \leq e^{-\frac{\sum_{i=1}^{k^*+1} c_i}{\lfloor B \rfloor + \zeta}} \cdot f(OPT_d) \\ & \quad = e^{-\frac{\Gamma(S_{k^*+1})}{\lfloor B \rfloor + \zeta}} \cdot f(OPT_d), \end{aligned} \tag{5}$$

Consequently, we get the relation between S_{k^*} and OPT_d as

$$f(S_{k^*+1}) \geq \left(1 - e^{-\Gamma(S_{k^*+1})/(\lfloor B \rfloor + \zeta)}\right) \cdot f(OPT_d). \tag{6}$$

Then we investigate the approximation by using the relation and the abandoned element z. By Observation 1 and definitions, it observes that $\Gamma(S_{k^*} \cup \{z\}) = \Gamma(S_{k^*}) + c' > \lfloor B \rfloor$, where $c' = \Gamma(z \mid S_{k^*})$. Putting it with (6) together gives

$$\begin{aligned} f(S_{k^*+1}) & \geq \left(1 - e^{-\frac{\Gamma(S_{k^*+1})}{\Gamma(S_{k^*}) + c' + \zeta}}\right) \cdot f(OPT_d) \\ & = \left(1 - e^{-1} exp\left(\frac{\zeta + c' - c_{k^*+1}}{\Gamma(S_{k^*}) + c' + \zeta}\right)\right) \cdot f(OPT_d). \end{aligned} \tag{7}$$

As S_{k^*+1} at least include one element, the expression of $exp(\cdot)$ is $(1 + o(1))$. Moreover, let $v^* \in V \setminus S_{k^*}$ be the element that has the largest function value. Observe that $f(v^*) \geq f(v_{k^*+1})$ and $f(S_{cc}) > f(S_{k^*})$. It gets $f(S_{cc}) + f(v^*) \geq f(S_{k^*}) + f(v^*) \geq f(S_{k^*+1})$. Putting them together gets $f(S_{cc}) + f(v^*) \geq (1 - o(1))(1 - 1/e) \cdot f(OPT_d)$, and therefore $\max_{Y \in \{S_{cc}, \{v^*\}\}} f(Y) \geq (1/2 - o(1))(1 - 1/e) \cdot f(OPT_d)$. □

6 Performance of the GGMA

In this section, we analyze the approximation behavior of the GGMA. The performance of the algorithm applying two different strategies is investigated separately.

6.1 Analysis of Using Strategy I

Theorem 5 implies that using Strategy I: $h(S) := \sum_{i \in S} \delta_i^2$, the GGMA also performs badly in the instances I_2, which was presented in the Section 5.1.

Theorem 5. *Given $\varepsilon \geq 1$, there exists an instance I_2 such that the GGMA applying $h := \sum_{i \in S} \delta_i^2$ fails to guarantee better than $(2/\varepsilon - 1/\varepsilon^2)-$approximation.*

6.2 Analysis of Using Strategy II

For Strategy II: $h(S) := \Gamma(S)$, let S_{cc} be the greedy solution constructed by the greedy strategy for the instance in the chance-constrained setting, and OPT_d be the optimal solution for the corresponding deterministic instance. In the i-th generation, the element v_i is selected by the algorithm, and its surrogate weight is denoted by c_i. Besides, the partial solution containing the first i item is denoted by $S_i \subseteq S_{cc}$. Then some useful greedy performance functions are defined to track the performance of the algorithm for the chance-constrained setting.

For a fixed $x \in [0, B]$, let i be the smallest greedy index so that $\Gamma(S_i) > x$. Then to track the performance of the greedy strategy, a continuous and monotone piecewise-linear function $g(x)$ is defined as $g(x) = f(S_{i-1}) + (x - \Gamma(S_{i-1}))\frac{f(v_i|S_{i-1})}{c_i}$, and $g(0) := 0$. Additionally, g' denotes the right derivative for g on the interval $[0, \Gamma(S_{cc}))$. Observe that g' is always non-negative as the objective value of the greedy solution does not decrease after including a new item. Besides, to track the performance of the greedy+Max when the greedy solution collects a set of cost x, we define the function $g_+(x) = g(x) + f(v \mid S_{i-1})$, where $v = \text{argmax}_{j \in V \setminus S_{i-1} : \Gamma(\{j\} \cup S_{i-1}) \leq B} f(j \mid S_{i-1})$.

After that, we consider the lower bound of the function g_+ in the specific interval. Following the definition of the fixed greedy index i, z_{max} denotes the element that has the largest dispersion in $OPT_d \setminus S_{i-1}$. Note that z_{max} is the first abounded element from OPT_d due to the chance constraint. Denote by S_{k^*} the partial solution. If S_{k^*} is obtained then z_{max} is removed. That implies z_{max} can be selected by the algorithm as the augmenting item for the set S_i where $0 \leq i \leq k^* - 1$. Therefore for $x \in [0, \Gamma(S_{k^*-1})]$, we define the greedy+Max performance lower bound as $g_1(x) = g(x) + f(z_{max} \mid S_{i-1})$, so that $g_1 \leq g_+$ for $x \in [0, \Gamma(S_{k^*-1})]$.

Now we investigate the relation between OPT_d, $g_1(x)$ and $g'(x)$ while $x \in [0, \Gamma(S_{k^*-1})]$ in Lemma 2 (proof in Appendix **??**).

Lemma 2. *For any $x \in [0, \Gamma(S_{k^*-1})]$, let i be the smallest greedy index so that $\Gamma(S_i) > x$. It holds that*

$$f(OPT_d) \leq g_1(x) + g'(x) \sum_{j \in OPT_d \setminus z_{max}} \Gamma(j|S_{i-1}).$$

Then we focus on the point $x = \Gamma(S_{k^*-1})$ and analyze the approximation behavior of the GGMA (Thereom 6) via Lemma 2.

Theorem 6. *The solution obtained by the GGMA applying $h := \Gamma(S)$ is a $(1/2 - o(1))-$approximation.*

Proof. Following Lemma 2 and applying $x = \Gamma(S_{k^*-1})$, it gives

$$\begin{aligned} f(OPT_d) &\le g_1(\Gamma(S_{k^*-1})) \\ &\quad + g'(\Gamma(S_{k^*-1})) \sum_{j \in OPT_d \setminus Z_{max}} \Gamma(j|S_{k^*-1}). \end{aligned} \tag{8}$$

Here the upper bound of the last term is given below. Recall that $|OPT_d| = \lfloor B \rfloor$. Let $\psi := \sqrt{Var[W(S_{k^*-1})]}$. It holds that

$$\begin{aligned} &\sum_{j \in OPT_d \setminus z_{max}} \Gamma(j|S_{k^*-1}) \\ &= \sum_{j \in OPT_d} \Gamma(j|S_{k^*-1}) - \Gamma(z_{max}|S_{k^*-1}) \\ &\le \sum_{j \in OPT_d} \Gamma(j|S_{k^*-1}) - \Gamma(z_{max}|S_{k^*}) \\ &= \lfloor B \rfloor + \eta - c'_{max}, \end{aligned} \tag{9}$$

where $\eta = \kappa_\alpha \sum_{j \in OPT_d} \left(\sqrt{Var[W(S_{k^*-1} \cup \{j\})]} - \psi \right)$ and $c'_{max} = \Gamma(z_{max}|S_{k^*})$. Consequently, we have

$$\begin{aligned} f(OPT_d) &\le g_1(\Gamma(S_{k^*-1})) \\ &\quad + g'(\Gamma(S_{k^*-1}))(\lfloor B \rfloor + \eta - c'_{max}). \end{aligned} \tag{10}$$

After that, we consider the value of $g_1(\Gamma(S_{k^*-1}))$ in (10). Recall that the surrogate weight of v_{k^*} is c_{k^*}. Then let $c^* := 1 + \kappa_\alpha \sqrt{\delta_{k^*}^2/3}$ and $\phi := \frac{c_{k^*} \cdot (\Gamma(S_{k^*-1}) + c^*)}{\Gamma(S_{k^*}) + \eta}$. Two possible cases for it are listed as follows.

Case 1. $g_1(\Gamma(S_{k^*-1})) \ge \frac{\phi}{c^* + \phi} \cdot f(OPT_d)$. Since $g_1(x) \le g_+(x)$ for $x \in [0, \Gamma(S_{k^*-1})]$, it directly holds $g_+(\Gamma(S_{k^*-1})) \ge \frac{\phi}{c^* + \phi} \cdot f(OPT_d)$.

Case 2. $g_1(\Gamma(S_{k^*-1})) < \frac{\phi}{c^* + \phi} \cdot f(OPT_d)$. For this case, we can prove that $g(\Gamma(S_{k^*})) \ge \frac{\phi}{c^* + \phi} \cdot f(OPT_d)$ as following. First of all, rearranging (10) gets

$$\begin{aligned} g'(\Gamma(S_{k^*-1})) &\ge \frac{f(OPT_d) - g_1(\Gamma(S_{k^*-1}))}{\lfloor B \rfloor + \eta - c'_{max}} \\ &\ge \frac{c^* \cdot f(OPT_d)}{(c^* + \phi)(\lfloor B \rfloor + \eta - c'_{max})}. \end{aligned} \tag{11}$$

Besides, let $g'_{min} := \arg\min_{i \in [1, k^*]} g'(\Gamma(S_{i-1}))$. Recall g' is non-negative and $g(0) = 0$, thus it holds that

$$g(x) \ge x \cdot g'_{min}, \tag{12}$$

for any $x \in [0, \Gamma(S_{k^*-1})]$. Now we show that $g'_{min} \ge \frac{f(v_{k^*}|S_{k^*-1})}{c^*}$. Considering the greedy strategy of the GGMA, it holds that $g'(\Gamma(S_{i-1})) = \frac{f(v_i|S_{i-1})}{c_i} \ge \frac{f(v_{k^*}|S_{i-1})}{\Gamma(v_{k^*}|S_{i-1})}$ in the i-th generation for $1 \le i \le k^*$. Observe $\Gamma(v_{k^*}|S_{i-1}) \le c^*$ and $f(v_{k^*}|S_{i-1}) \ge f(v_{k^*}|S_{k^*-1})$ for $i \le k^*$ as $S_{i-1} \subseteq S_{k^*-1}$ (recall (1)). Therefore we have $\frac{f(v_{k^*}|S_{i-1})}{\Gamma(v_{k^*}|S_{i-1})} \ge \frac{f(v_{k^*}|S_{k^*-1})}{c^*}$. Putting them together yields $g'_{min} \ge \frac{f(v_{k^*}|S_{k^*-1})}{c^*}$. For any $x \in [0, \Gamma(S_{k^*-1})]$, therefore it holds that

$$g(x) \ge x \cdot \frac{f(v_{k^*}|S_{k^*-1})}{c^*}. \tag{13}$$

Then applying $x = \Gamma(S_{k^*-1})$ to (13), it gets

$$g(\Gamma(S_{k^*-1})) \ge \Gamma(S_{k^*-1}) \cdot \frac{f(v_{k^*} \mid S_{k^*-1})}{c^*} \tag{14}$$

Besides, since $g'(\Gamma(S_{k^*-1})) = \frac{f(v_{k^*}|S_{k^*-1})}{c_{k^*}}$, rearranging (11) gets $\frac{f(v_{k^*}|S_{k^*-1})}{c^*} \ge \frac{c_{k^*} \cdot f(OPT_d)}{(c^* + \phi)(\lfloor B \rfloor + \eta - c'_{max})}$. Putting them together, we have $g(\Gamma(S_{k^*-1})) \ge \frac{f(OPT_d) \cdot \Gamma(S_{k^*-1}) \cdot c_{k^*}}{(c^* + \phi)\left(\lfloor B \rfloor + \eta - c'_{max}\right)}$.

Now we can derive a lower bound for the objective value of the set S_{k^*}. Recall that v_{k^*} is the next added element for the solution S_{k^*-1}. Thus $f(S_{k^*})$ is at least

$$\begin{aligned} g(\Gamma(S_{k^*})) &= g(\Gamma(S_{k^*-1})) + c_{k^*} g'(\Gamma(S_{k^*-1})) \\ &\ge \frac{c_{k^*} \cdot (\Gamma(S_{k^*-1}) + c^*)}{(c^* + \phi)(\lfloor B \rfloor + \eta - c'_{max})} \cdot f(OPT_d). \end{aligned} \tag{15}$$

Furthermore, by Observation 1 it yields that $\Gamma(S_{k^*-1} \cup \{v_{k^*}, z_{max}\}) = \Gamma(S_{k^*}) + c'_{max} \ge \lfloor B \rfloor$. Put them together gets

$$\begin{aligned} g(\Gamma(S_{k^*})) &\ge \frac{c_{k^*} \cdot (\Gamma(S_{k^*-1}) + c^*)}{(c^* + \phi)(\lfloor B \rfloor + \eta - c'_{max})} \cdot f(OPT_d) \\ &\ge \frac{c_{k^*} \cdot (\Gamma(S_{k^*-1}) + c^*)}{(c^* + \phi)(\Gamma(S_{k^*}) + \eta)} \cdot f(OPT_d) \\ &= \frac{\phi}{c^* + \phi} \cdot f(OPT_d). \end{aligned} \tag{16}$$

Therefore the GGMA achieves a $\frac{\phi}{c^* + \phi}$-approximation. Since at least one element is included in the greedy solution, the expression of $\frac{\phi}{c^* + \phi}$ is $(1/2 - o(1))$. Additionally, as the objective value of the augmented output solution T is no worse than $g_+(\Gamma(S_{k^*}))$, it holds $f(T) \ge (1/2 - o(1)) \cdot f(OPT_d)$. □

7 Experiments

In this section, we regard the GA as the baseline algorithm and evaluate the experimental performance of other algorithms (namely the GGA and the GGMA) on two significant submodular optimization problems such as the maximum coverage problem (MCP) and the influence maximum problem (IMP) with chance constraint. Following the specific setting described in Section 2, the expectation of each element's weight is set as 1, and the dispersion value of each item is different.

7.1 The Maximum Coverage Problem

The first submodular problem is the maximum coverage problem [5, 9]. We consider a chance-constrained version of the MCP based on the graph. Given an undirected graph $G = (V, E)$, we denote the degree of the node v_i by $D(v_i)$, and the number of all nodes of $V' \subseteq V$ and their neighbors in G by the objective function $N(V')$. The MCP aims to find a subset V' so that $N(V')$ is maximized under the constraint. Moreover, given a linear cost function $c : V \to \mathbb{R}^+$, the problem under chance constraint is formulated as

$$\underset{V' \subseteq V}{\operatorname{argmax}}\ N(V')\ s.t.\ Pr[c(V') > B] \le \alpha. \tag{17}$$

The graph used in the instances are frb30-15-01 (450 nodes and 17827 edges) and frb35-17-01 (595 nodes and 27 856 edges) [22]. In terms of settings, for each node v_i, the cost a_i is 1 but the value of the dispersion δ_i is set by two different methods. The first method is that δ_i is independently uniformly at random sampled from $[0, 1]$. To analyze our experiments more rigorously, we independently randomly sample the value of dispersion five times for each graph. Moreover, we also consider δ_i is associated with the degree of the

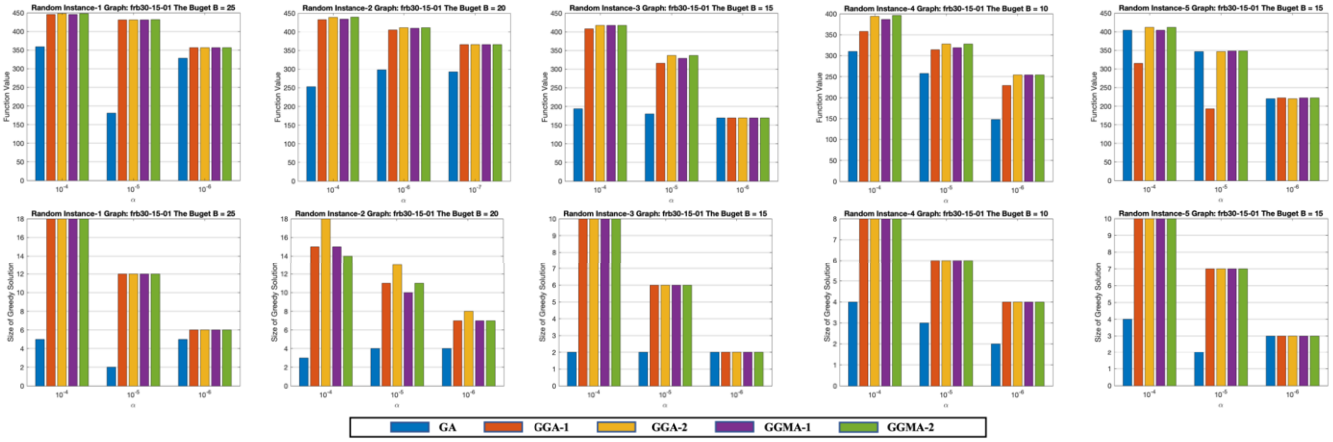

Figure 1. $N(V')$ (top) and $|V'|$ (bottom) for the graph frb30-15-01 with different budgets when δ is randomly sampled from the uniform distribution.

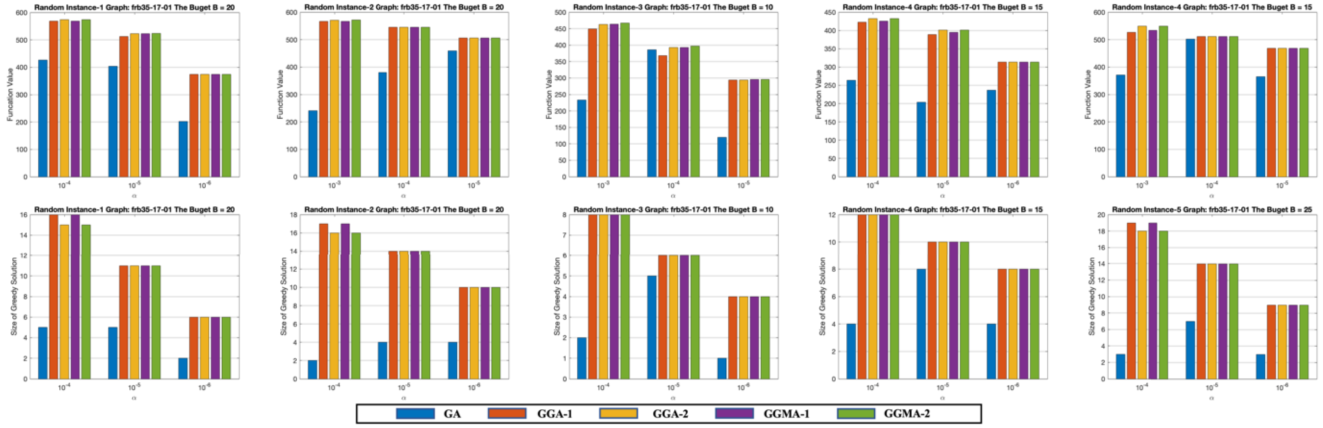

Figure 2. $N(V')$ (top) and $|V'|$ (bottom) for the graph frb35-17-01 with different budgets when δ is randomly sampled from the uniform distribution.

node, which can be expressed as $\delta_i = D(v_i)/\sum_{v \in V} D(v)$. Furthermore, we investigate all combinations of $\alpha \in \{10^{-4}, 10^{-5}, 10^{-6}\}$ and $B \in \{10, 15, 20, 25\}$ for the experimental investigation of the algorithms. The performance of the algorithms is measured in terms of the function value $N(V')$.

The experimental results are shown in figures 1, 2 and 3, which indicate that for the instances with the same budget, both the function value and the number of nodes of the solution by the algorithms GGA and GGMA decline as α increasing. They also show that the performance of the GA is worse than GGA and GGMA using strategy II among most instances. It also can be found that GA collects fewer items before reaching the budget, which matches our theoretical analysis. Besides, for the same strategy applied to the different algorithms, the GGMA slightly outperforms the GGA while applying strategy I to the instances with a lower budget. Additionally, the performance of the GGMA using strategy II is comparable to the GGA.

In terms of strategies, we observe from figures that using strategy II can improve the performance of all algorithms by strategy I. More precisely, the algorithms with strategy I can output a solution that includes more nodes but is with a lower function value, among most instances. It is noticeable that the GGMA with strategy II can obtain high-quality solutions for these instances.

7.2 *The Influence Maximization Problem*

We now study the influence maximization problem [28, 19, 11]. The IMP aims to identify the set of users, who are the most influential in a large-scale social network.

The goal of the IMP is to maximize the spread of influence over a given social network, i.e., a graph of interactions within a group of users [8]. This section presents the experimental analysis of some chance-constrained IMP instances.

Let a directed graph be $G = (V, E)$ to represent a social network, in which each node $v_i \in V$ corresponds to a user, and the probability $p_{i,j}$ of the edge in E represents the strength of the influence between a pair of users v_i and v_j. The IMP aims to find a subset $X \subseteq V$ such that the expected number of nodes $E[I(X)]$ (the objective function of the problem) activated by propagating from X is maximized subject to the constraints. Given a linear cost function $c : V \to \mathbb{R}^+$ and a budget B, the chance constraint version of the IMP is formulated as

$$\underset{X \subseteq V}{\operatorname{argmax}}\ E[I(X)]\ s.t.\ Pr[c(X) > B] \leq \alpha. \tag{18}$$

The dataset *Social circles: Facebook* consists of friends lists collected from a social networking service, which includes 4,039 nodes and 176,468 edges [12]. We transform the instances in this dataset to chance-constrained IMP instances. For each node v_i, its expected cost is set as 1, and the dispersion δ_i is independently and uniformly sampled from $[0, 1]$. The algorithms are evaluated for all pairs of budgets $B \in \{20, 50, 100, 150\}$ and tolerate probabilities

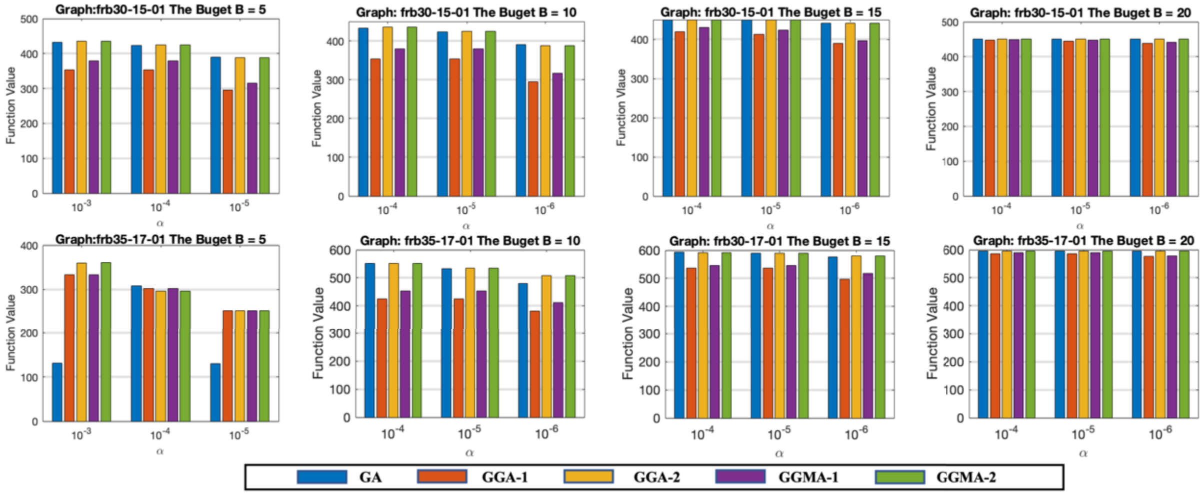

Figure 3. $N(V')$ for the graphs frb30-15-01 (top) and frb35-17-01 (bottom) with different budgets when δ is based on the degree.

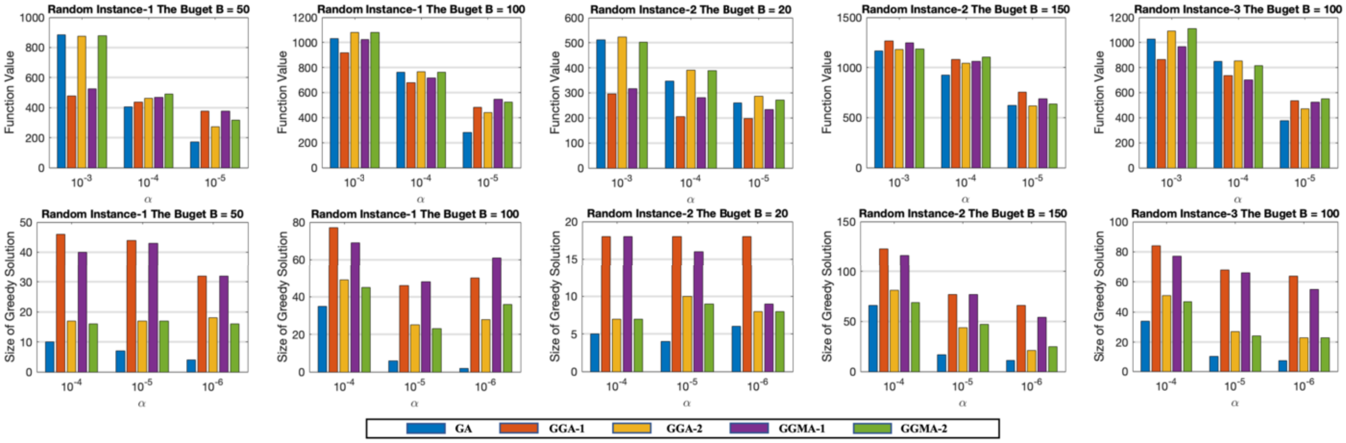

Figure 4. $E[I(X)]$ (top) and $|X|$ (bottom) for IMP with different budgets when δ is randomly sampled from the uniform distribution.

$\alpha \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. We use the function value $E[I(X)]$ to evaluate the performance of algorithms, and independently sample the value of dispersion three times to analyze our experiments more rigorously.

Figure 4 clearly shows that the GA collects fewer nodes than other algorithms and performs unwell when α is small. For the same budget instances, the function value obtained sharply decreases with the increasing value of α. This phenomenon is common among those algorithms. Moreover, we observe that the GGMA includes fewer nodes than GGA but obtains higher function values in most instances.

In terms of strategies, the results demonstrate that the algorithms with strategy I is significantly worse in some instances than them with strategy II, even worse than the GA although it collects more elements. On the other hand, the algorithms applying strategy II can fix it in most instances, which coincides with our theoretical analysis. In addition, the GGMA beats the GGA in terms of the quality of the output solution in most instances.

8 Conclusion

The paper studied a chance-constrained submodular optimization problem with variable uncertainties and investigated the performance of the GA, the GGA, and the GGMA on it. In the setting, the weights of elements are sampled from distributions with the same expectation but varied dispersion values. We found that the GA does not perform well even in some linear instances. Besides the GGA and the GGMA respectively can achieve guarantee a $(1/2 - o(1))(1 - 1/e)-$approximation and a $(1/2 - o(1))-$approximation of the optimal solution for a deterministic setting. Additionally, the experimental results showed that the GGMA using the surrogate weight based on the one-sided Chebyshev's inequality beats other algorithms in some instances of the MCP and the IMP which are typical submodular problems.

The future work is to broaden the exploration of chance-constrained submodular problems with variable weights and uncertainties, and potentially different distributions. These subsequent studies will be both challenging and engaging, with the aim of yielding more meaningful insights to enhance our comprehension of the problem.

Acknowledgements

This work has been supported by the Australian Research Council (ARC) through grant FT200100536, the Hunan Provincial Natural Science Foundation of China through grant 2021JJ40791, and the Open Project of Xiangjiang Laboratory (No.22XJ03005).

References

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

[2] Junjie Chen and Takanori Maehara, 'Chance-constrained submodular knapsack problem', in *International Computing and Combinatorics Conference*, pp. 103–114. Springer, (2019).

[3] Abhimanyu Das and David Kempe, 'Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection', in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1057–1064, (2011).

[4] Benjamin Doerr, Carola Doerr, Aneta Neumann, Frank Neumann, and Andrew Sutton, 'Optimization of chance-constrained submodular functions', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1460–1467, (2020).

[5] Uriel Feige, 'A threshold of ln n for approximating set cover', *Journal of the ACM (JACM)*, **45**(4), 634–652, (1998).

[6] Tobias Friedrich, Andreas Göbel, Frank Neumann, Francesco Quinzan, and Ralf Rothenberger, 'Greedy maximization of functions with bounded curvature under partition matroid constraints', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2272–2279, (2019).

[7] Kakuzo Iwamura and Baoding Liu, 'A genetic algorithm for chance constrained programming', *Journal of Information and Optimization sciences*, **17**(2), 409–422, (1996).

[8] David Kempe, Jon Kleinberg, and Éva Tardos, 'Maximizing the spread of influence through a social network', in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, (2003).

[9] Samir Khuller, Anna Moss, and Joseph Seffi Naor, 'The budgeted maximum coverage problem', *Information processing letters*, **70**(1), 39–45, (1999).

[10] Andreas Krause and Daniel Golovin, 'Submodular function maximization.', *Tractability*, **3**, 71–104, (2014).

[11] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance, 'Cost-effective outbreak detection in networks', in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, (2007).

[12] Jure Leskovec and Julian Mcauley, 'Learning to discover social circles in ego networks', *Advances in neural information processing systems*, **25**, (2012).

[13] Bruce L Miller and Harvey M Wagner, 'Chance constrained programming with joint constraints', *Operations Research*, **13**(6), 930–945, (1965).

[14] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher, 'An analysis of approximations for maximizing submodular set functions—i', *Mathematical programming*, **14**(1), 265–294, (1978).

[15] Aneta Neumann and Frank Neumann, 'Optimising monotone chance-constrained submodular functions using evolutionary multi-objective algorithms', in *International Conference on Parallel Problem Solving from Nature*, pp. 404–417. Springer, (2020).

[16] Frank Neumann and Andrew M Sutton, '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*, pp. 147–153, (2019).

[17] Frank Neumann and Carsten Witt, 'Runtime analysis of single-and multi-objective evolutionary algorithms for chance constrained optimization problems with normally distributed random variables', in *31st International Joint Conference on Artificial Intelligence*, pp. 4800–4806. International Joint Conferences on Artificial Intelligence Organization, (2022).

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

[19] Chao Qian, Jing-Cheng Shi, Yang Yu, and Ke Tang, 'On subset selection with general cost constraints.', in *IJCAI*, volume 17, pp. 2613–2619, (2017).

[20] Chao Qian, Jing-Cheng Shi, Yang Yu, Ke Tang, and Zhi-Hua Zhou, 'Subset selection under noise', *Advances in neural information processing systems*, **30**, (2017).

[21] Vahid Roostapour, Aneta Neumann, Frank Neumann, and Tobias Friedrich, 'Pareto optimization for subset selection with dynamic cost constraints', *Artificial Intelligence*, **302**, 103597, (2022).

[22] Ryan A. Rossi and Nesreen K. Ahmed, 'The network data repository with interactive graph analytics and visualization', in *AAAI*, (2015).

[23] Feng Shi, Xiankun Yan, and Frank Neumann, 'Runtime analysis of simple evolutionary algorithms for the chance-constrained makespan scheduling problem', in *International Conference on Parallel Problem Solving from Nature*, pp. 526–541. Springer, (2022).

[24] Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, and Frank Neumann, 'Evolutionary algorithms for the chance-constrained knapsack problem', in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 338–346, (2019).

[25] Yue Xie, Aneta Neumann, and Frank Neumann, 'Specific single- and multi-objective evolutionary algorithms for the chance-constrained knapsack problem', in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 271–279, (2020).

[26] Yue Xie, Aneta Neumann, Frank Neumann, and Andrew M Sutton, 'Runtime analysis of rls and the (1+ 1) ea for the chance-constrained knapsack problem with correlated uniform weights', in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1187–1194, (2021).

[27] Grigory Yaroslavtsev, Samson Zhou, and Dmitrii Avdiukhin, '"bring your own greedy"+ max: near-optimal 1/2-approximations for submodular knapsack', in *International Conference on Artificial Intelligence and Statistics*, pp. 3263–3274. PMLR, (2020).

[28] Haifeng Zhang and Yevgeniy Vorobeychik, 'Submodular optimization with routing constraints', in *Proceedings of the AAAI conference on artificial intelligence*, volume 30, (2016).