Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Efficient Algorithms with Performance Guarantees for the Stochastic Multiple-Choice Knapsack Problem

Long Tran–Thanh[*] and **Yingce Xia**[†] and **Tao Qin**[‡] and **Nicholas R. Jennings**[§]

Abstract

We study the stochastic multiple-choice knapsack problem, where a set of K items, whose value and weight are random variables, arrive to the system at each time step, and a decision maker has to choose at most one item to put into the knapsack without exceeding its capacity. The goal of the decision-maker is to maximise the total expected value of chosen items with respect to the knapsack capacity and a finite time horizon. We provide the first comprehensive theoretical analysis of the problem. In particular, we propose OPT-S-MCKP, the first algorithm that achieves optimality when the value-weight distributions are known. This algorithm also enjoys $\tilde{O}(\sqrt{T})$ performance loss, where T is the finite time horizon, in the unknown value-weight distributions scenario. We also further develop two novel approximation methods, FR-S-MCKP and G-S-MCKP, and we prove that FR-S-MCKP achieves $\tilde{O}(\sqrt{T})$ performance loss in both known and unknown value-weight distributions cases, while enjoying polynomial computational complexity per time step. On the other hand, G-S-MCKP does not have theoretical guarantees, but it still provides good performance in practice with linear running time.

1 Introduction

The class of knapsack problems is a fundamental set of NP-hard combinatorial optimisation problems that are widely used in many areas, ranging from finance and management science, to computer science and artificial intelligence. The standard (or 0-1) knapsack problem consists of a knapsack with capacity C, and a set of items, each of which has certain value and weight. The goal of the decision maker (i.e., the optimiser) is to find a set of items with maximum total value such that the sum of the items' weight does not exceed the capacity C of the knapsack. Other popular variants of the knapsack require different constraints, such as: (i) choosing multiple copies of a single item is allowed (bounded/unbounded knapsack); (ii) choosing one from each subset of items (multiple-choice knapsack); or (iii) putting items into multiple knapsacks (multi-knapsack) (for further details of the knapsack problems, see, e.g., [Kellerer *et al.*, 2004]).

In the classical (or *offline*) setting of the knapsack problems, the items are typically available for the decision maker from the beginning of the process, with deterministic and known values and weights. However, a number of researchers are recently focussing on the *stochastic* version [Lueker, 1995; Babaioff *et al.*, 2007; Tran-Thanh *et al.*, 2012], where a set of new items arrive to the system at each time step in an online manner, and the value and weight of these items are typically randomly drawn from a distribution. The decision maker then chooses (a subset of) items to place into the knapsack without exceeding its capacity, while disposing of the others (which cannot be reused in the future). This stochastic setting has many applications, especially in the field of artificial intelligence and its related areas, such as stochastic planning, sequential decision theory, and online machine learning [Cohn and Mit, 1998; Lu *et al.*, 1999; Benoist *et al.*, 2001; Tran-Thanh *et al.*, 2012], as problems within these areas typically consist of real-time processes, where the decisions have to be made sequentially at each time step. Despite the important role of the stochastic knapsacks in many research areas, and the fact that their offline counterpart have been thoroughly investigated for many years, only a small number of efficient algorithms, that require low computational costs and enjoy good theoretical performance guarantees, have been proposed within this area. In more detail, while researchers have devised approximation algorithms, that are both theoretically and practically efficient, for the stochastic 0-1 [Lueker, 1995; Kakade *et al.*, 2007] and bounded/unbounded knapsacks [Ding *et al.*, 2013; Tran-Thanh *et al.*, 2014], similar efficient algorithms for other stochastic knapsacks have not yet been proposed.

Against this background, this paper aims to fill the gap by introducing new efficient algorithms for the stochastic multiple-choice knapsack problem (S-MCKP), an important variant of the stochastic knapsack, which occurs in many

[*]Corresponding author. University of Southampton, UK. Email: l.tran-thanh@soton.ac.uk.

[†]University of Science and Technology of China. Email: yingce.xia@gmail.com.

[‡]Microsoft Research. Email: taoqin@microsoft.com.

[§]University of Southampton, UK. Email: nrj@ecs.soton.ac.uk.

sequential decision making problems. In particular, in S-MCKP, a set of K items, whose value and weight are randomly drawn from stationary value-weight distributions, arrive to the system at each time step, and a decision maker has to choose at most one item to put into the knapsack, in order to maximise the total *expected* value of the chosen items, *before* the true value and weight of each item are revealed to her. This problem is motivated by many real-world applications. For example, consider the search and rescue domain in robotics, where an autonomous rescue robot has to decide at each round (i.e., time step) where to go. To reach different locations, the robot has to consume a certain amount of energy (i.e., weight) from its limited battery (i.e., capacity), and receives a certain utility (i.e., value) when it executes the task. Since it can have access to the observations of the other first responders and searchers, it can also evaluate the cost and utility of other actions it did not choose (i.e., all the value-weight pairs are revealed) *at the end* of each round. The goal of the robot is then to maximise its total utility over a finite time horizon, with respect to its limited battery.

While the S-MCKP has been studied as a special case of a more generic model [Badanidiyuru *et al.*, 2013], or with simplified settings [Zhou and Naroditskiy, 2008], state-of-the-art algorithms are still computationally expensive and do not have good performance guarantees, and thus, cannot be used in real-world applications (see Section 2 for more details). In addition, other existing stochastic knapsack algorithms, that are designed for different settings, are not suitable for efficiently handling the multiple-choice constraint, and thus, they will not provide good performance in S-MCKP either. To this end, our main goal is to develop fast algorithms that can provide good theoretical performance guarantees for the S-MCKP. In particular, we consider two settings that together cover all the possible scenarios, namely: (i) when the value-weight distribution functions are known; and (ii) when this distribution functions are not known *a priori*. Given these two settings, we first introduce OPT-S-MCKP, an algorithm which is provably optimal (i.e., provides the best expected performance) for the case of known value-weight distributions. We also show that its performance loss (i.e., the difference between an algorithm's performance to that of the best possible) in the case of having unknown value-weight distributions is at most $\tilde{O}(\sqrt{T})$ where T is the finite time horizon, and the logarithmic terms are hidden in the $\tilde{O}$ notation. Note that the sub-linear (e.g, squared root) performance loss implies that the average performance loss per time step is converging to 0 as T tends to infinity. Thus, the behaviour of the algorithm with a sub-linear performance loss converges to that of an optimal algorithm. However, as S-MCKP is NP-hard, this optimal algorithm cannot be computationally efficient. To overcome this issue, we additionally propose two novel approximation algorithms, FR-S-MCKP; and G-S-MCKP, respectively. These algorithms represent two dominant classes of approximation techniques within the knapsack literature. In particular, the former follows a fractional relaxation approximation approach, and enjoys a $\mathrm{poly}(K)$ computational cost per time step (K is the number of items per time step), while the latter applies a greedy approximation method, and achieves linear computational complexity. In addition, we also prove that FR-S-MCKP provably achieves performance loss of at most $\tilde{O}(\sqrt{T})$ for both cases of known and unknown value-weight distributions. In contrast, while similar performance guarantees do not exist for G-S-MCKP, we demonstrate through extensive numerical evaluation that it can achieve similar performance, compared to that of the other two methods, with significantly lower computational cost. In summary, we extend the state of the art as follows:

- We provide the first comprehensive theoretical analysis for a stochastic knapsack problem, S-MCKP, that is widely used in many real-world applications. In particular, we propose OPT-S-MCKP, the first method that is optimal when the value-weight distributions are known, and can achieve performance loss guarantee of $\tilde{O}(\sqrt{T})$ when the distributions are not given *a priori*.
- We also propose two novel approximation algorithms, R-S-MCKP, and G-S-MCKP, that can achieve efficient performance with low computational costs, and we show that FR-S-MCKP can provably achieve $\tilde{O}(\sqrt{T})$ performance loss guarantee for both cases of known and unknown distributions.

In the remainder of the paper, we first provide a review of the related work (Section 2). We then describe S-MCKP (Section 3) and the algorithms in more detail (Sections 4 and 5). We also provide the theoretical analysis (Theorems 1, 2, and 3) and numerical evaluation of the algorithms (Section 6).

2 Related Work

[Lueker, 1995] was the first to investigate knapsack problems in the stochastic setting. In particular, he proposed a greedy approach to approximate the stochastic 0-1 knapsack problem, and he proved that the algorithm can guarantee a performance loss of at most $O(\ln T)$. His results were later extended by [Kakade *et al.*, 2007], who described an approximation technique with $\tilde{O}(\sqrt{T})$ performance loss guarantee for the online 0-1 knapsack problem with adversarial values and weights (i.e., here the sequence of values and weights is not randomly drawn from a distribution, but is set *a priori* by an adversary). These approaches, however, are specifically designed for the 0-1 knapsack, and thus, cannot be applied to our settings, as they heavily rely on the fact that the decision space at each time step is binary.

More similar to our paper are the works on bounded and unbounded stochastic knapsacks (which are in fact special cases of the multiple-choice version, with the time horizon set to be infinite). More specifically, [Tran-Thanh *et al.*, 2010; 2012; Ding *et al.*, 2013] investigated the unbounded setting, and proposed algorithms that can achieve $O(\ln T)$ performance loss, compared to that of the best possible. On the other hand, existing algorithms for bounded (i.e. there is a limit on the number of copies we can choose from each single item) knapsack are less efficient. In particular, [Tran-Thanh *et al.*, 2014] proposed an algorithm with $O(T^{2/3})$ performance loss guarantee, while the algorithm developed by [Ho and Wortman-Vaughan, 2012] can achieve a performance approximation with a comparative ratio (i.e., the performance

loss is linear). Nevertheless, the abovementioned algorithms, both for unbounded and bounded knapsacks, cannot be applied to our settings as they are designed for special cases of our model, that does take the finite time horizon T into account. Thus, they may put into a knapsack a sequence of items that contains significantly larger items than T, which is the maximal number of items we can put into the knapsack (as at most one item can be chosen per time step). More recently, [Badanidiyuru *et al.*, 2013] investigated the stochastic multi-dimensional knapsack, that can be regarded as a generalisation of many knapsack problems, and proposed two approximation algorithms with guaranteed performance losses. However, these algorithms are designed for very generic settings, and thus, are computationally very involved, with inefficient performance guarantees, compared to our results. In particular, their results guarantee a $\tilde{O}(\sqrt{\mathrm{OPT}})$ upper bound on the loss, where OPT is the performance of the optimal solution, which is typically significantly larger than $\tilde{O}(\sqrt{T})$. Apart from this, S-MCKP was also studied by [Zhou and Naroditskiy, 2008] with a modified setting, where all the value-weight pairs are revealed before each decision. However, their algorithm does not have any performance guarantees.

It is worth mentioning that a special case of our setting, where all the weights are set to be equal and deterministic, is a well studied topic in the online machine learning literature, known as the problem of learning with expert advices [Cesa-Bianchi *et al.*, 1997; Cesa-Bianchi and Lugosi, 2006]. Within this area, state-of-the-art algorithms typically enjoy a constant performance loss, compared to that of the *best fixed* solution on hindsight. However, as these algorithms do not take the weights into account, they are not suitable to tackle S-MCKP, where the role of the weights are essential. More recently, [Amin *et al.*, 2015] introduced weights (as advice costs) into this domain of learning with expert advice. However, their model only considers fixed and deterministic weights, with a per time step capacity, which is refilled after each time step. Given these differences in the model, their proposed method does not fit into our setting, and thus, will not be suitable for providing efficient performance.

3 The S-MCKP Model

In this section we describe the stochastic multiple-choice knapsack problem (S-MCKP) in more detail. The S-MCKP model consists of a knapsack with capacity C. At each time step $t \in \{1, \dots, T\}$, we have to choose one from a set of K items $\mathbf{K}_t = \{(v_t(k), w_t(k)) | 1 \leq k \leq K\}$ where $v_t(k)$ and $w_t(k)$ are the value and the weight of the k^{th} item from $\mathbf{K}_t$, respectively. Let $\mathbf{V}_t = \{v_t(k)\}_{k=1}^K$ and $\mathbf{W}_t = \{w_t(k)\}_{k=1}^K$ denote the vectors of values and weights of the items at time step t, respectively. In our setting, we assume that these value and weight vectors are drawn from stationary value-weight distributions. Moreover, let $\mathbf{D}_V$ and $\mathbf{D}_W$ denote the marginal distribution of value vector $\mathbf{V}_t$ and weight vector $\mathbf{W}_t$, respectively. For now, we assume that $\mathbf{D}_W$ is discrete, that is, $w_t(k)$ are discrete values[1]. We also assume that there exist $V, W > 0$ such that for each $1 \leq k \leq K$ and $1 \leq t \leq T$, $v_t(k) \leq V$ and $w_t(k) \leq W$ with probability 1. Now, let $p_k(w)$ denote the probability that $w_k(t) = w$. In addition, let $\mu_k = \mathbb{E}[v_t(k)]$ denote the expected value we can get if we choose the k^{th} item of $\mathbf{K}_t$. Since $\mathbf{K}_t$ is drawn from a stationary joint value-weight distribution, $\mathbb{E}[v_t(k)]$ remains the same for all t and thus, we can leave the time index out from μ_k. Let $\mu = \{\mu_k\}_{k=1}^K$ be the vector of these expected values.

In our model, the concrete value and weight of each item in $\mathbf{K}_t$ is only revealed after we have chosen an item from $\mathbf{K}_t$ (these values are also revealed if we decide not to choose any items). Our goal is to choose *at most one* item from each $\mathbf{K}_t$ and put it into the knapsack such that the total weight cannot exceed the capacity C and the expected total value of the chosen items is maximised. More formally, let $\mathbf{x}_t \in \{0,1\}^K$ denote a binary vector that represents our item choice at time step t. That is, $\sum_{k=1}^K x_t(k) \leq 1$. Our objective is to find a sequence of $\mathbf{x}_t$ for $t = 1, 2, \dots, T$ such that we achieve the optimal solution for the following constrained problem:

$$\max \mathbb{E}\Big[\sum_{t=1}^{T} \mathbf{x}_t \mathbf{V}_t\Big] \quad \text{s.t.} \quad \sum_{t=1}^{T} \mathbf{x}_t \mathbf{W}_t \leq C \quad \text{a.s.} \tag{1}$$

In what follows, we will discuss two cases, that covers all the possible scenarios of the problem, in more detail, namely: (i) we know in advance the joint value-weight distribution of the items (Section 4); and (ii) we do not have prior knowledge about this joint value-weight distribution (Section 5).

4 S-MCKP with Known Distributions

In this section, we assume that we have accurate prior knowledge about the joint value-weight distribution, and thus, both $\mathbf{D}_V$ and $\mathbf{D}_W$ are known in advance. To tackle this version of S-MCKP, we first propose an optimal solution (Section 4.1), which is computationally involved. To overcome the computational issues, we propose two near-optimal heuristics, each of which is taken from a popular class of approximation algorithms within the knapsack literature, that are computationally efficient (Sections 4.2 and 4.3).

4.1 An Optimal Solution

We first provide an optimal solution of the S-MCKP model. To do so, we introduce the following terms. Let $G^*(c,t)$ denote the maximal expected total value that we can achieve from time step t, with remaining budget c. Note that what we want to calculate is $G^*(C,1)$. In addition, let $G^*(c,t,k)$ denote the maximal expected total value we can achieve if we choose the k^{th} item from $\mathbf{K}_t$, with remaining budget c. To formalise the fact that we can decide not to choose any of the items at time step t, we introduce a new item indexed by $k = 0$ such that it always returns value and weight 0. This implies that $\mu_0 = w_0 = 0$. Now we turn to the description of the Bellman equations. Note that when $t \geq T + 1$ (i.e., we exceed the time horizon), for any $c > 0$, we have:

$$\forall t \geq T + 1 : \quad G^*(c,t) = G^*(c,t,k) = 0 \tag{2}$$

[1]This assumption can be easily relaxed to real-value weights. In fact, the main purpose of having this assumption is that it is easier to describe the main idea of OPT-S-MCKP on discrete weights, while FR-S-MCKP and G-S-MCKP can work with real-value weights without modification. To adopt OPT-S-MCKP to the case of real-value weights, we only have to replace the Bellman equations described in Section 4.1 with their differential equation counterparts.

where $0 \leq k \leq K$. In fact, this indicates that when we reach the time horizon, no additional item (and thus, value) can be added to the knapsack. Furthermore, for any $c \leq 0$, $1 \leq t \leq T$, and $0 \leq k \leq K$, we have:

$$\forall c \leq 0 : G^*(c,t) = G^*(c,t,k) = 0 \tag{3}$$

That is, we cannot achieve any improvements if we exceed the capacity. We now have the following recursion. For each $1 \leq t \leq T$, and $0 \leq k \leq K$, we have

$$G^*(c,t,k) = \mu_k + \sum_{w=1}^{W} p_k(w) G^*(c-w, t+1) \tag{4}$$

This implies that $G^*(c,t) = \max_{0 \leq k \leq K} G^*(c,t,k)$. We can also identify the item which we have to choose from $\mathbf{K}_t$ (including the fake item $k=0$) in the optimal solution. In particular, let $k^*(c,t)$ denote this optimal item at time step t with the remaining capacity is c. We have:

$$k^*(c,t) = \arg \max_{0 \leq k \leq K} G^*(c,t,k) \tag{5}$$

Note that $k^*(c, T+1) = 0$ for any c. It is known that the solution of this set of Bellman equations (or dynamic programming) provides the optimal solution of S-MCKP [Kellerer *et al.*, 2004]. Indeed, with this recursion, we can identify $k^*(C,1)$, which is the first item of the optimal solution. Given this, the optimal algorithm, OPT-S-MCKP, is as follows:

Initialisation: We set $t=1$ and $c_t = C$, respectively.

Solving the equations: We solve the Bellman equations described in Eqs (2) - (5) for each t and c_t, from which we obtain $k^*(c_t, t)$, the optimal item to be chosen at each time step t and residual capacity c_t.

Iterative steps:

1. We put $k^*(c_t,t)$ into the knapsack and observe its value and weight. Let $w_{k^*(c_t,t)}$ denote the latter.
2. We set $c_{t+1} = c_t - w_{k^*(c_t,t)}$ and $t = t+1$. We repeat Step 1 with the new time and capacity values.

This algorithm, however, is computationally involved, as we have to solve a set of $O(KTC)$ Bellman equations. This will lead to heavy computational cost in case of large T and C, which is typical in many real-world applications(see our experiments for more details). Given this, we next discuss two computationally more efficient approximation algorithms (i.e., with polynomial computation cost) for the S-MCKP.

4.2 Fractional Relaxation-based Approximation

We start with the description of FR-S-MCKP, a fractional relaxation-based approximation method for S-MCKP. In particular, the approach relies on the following idea. One way to overcome the computational complexity of OPT-S-MCKP is to relax the original problem described in Eq (1). In particular, let $\nu_k = \mathbb{E}[w_k]$ denote the expected weight of item k. Let vector $\nu = \{\nu_k\}_{k=1}^K$ denote by the set of these expected weights. Similarly to the case of OPT-S-MCKP, we also add a dummy item with index $k=0$, fixed value 0, and fixed weight 0, respectively. We consider the following fractional relaxation of the S-MCKP:

$$\max \mathbb{E}\Big[\sum_{t=1}^{T} \mathbf{x}_t \mathbf{V}_t\Big] = \max \sum_{t=1}^{T} \mathbf{x}_t \mu \quad \text{s.t.} \quad \sum_{t=1}^{T} \mathbf{x}_t \nu \leq C$$

$$\forall t, k: \quad 0 \leq x_t(k) \leq 1 \ \text{ and } \ \sum_{k=0}^{K} x_t(k) \leq 1 \tag{6}$$

That is, we allow $x_t(k)$ to be fractional (and not binary as in the original S-MCKP). In addition, we also relax the capacity constraint so that instead of restricting the total sum of weights has to be smaller than C for almost surely (as it is the case in the original problem), we only require the average total weight to be smaller than the capacity of the knapsack. This relaxation enjoys the advantage of having a polynomial (in the number of items) computational cost to calculate its optimal solution [Bagchi *et al.*, 1996]. Suppose that $\{\mathbf{x}_t^*\}_{t=1}^T = \{\{x_t^*(k)\}_{k=0}^K\}_{t=1}^T$ denote the optimal solution of the abovementioned relaxed problem. Since by modifying the contribution of the dummy item to this solution does not violate the capacity constraint at all, we can increase $x_t^*(0)$ until we exceed 1. That is, we set the new $x_t^*(0) := 1 - \sum_{k=1}^K x_t^*(k)$. By doing so, we can always guarantee that the sum of the elements within the optimal solution $\{\mathbf{x}_t^*\}_{t=1}^T$ is always T (this will play an important role in the proofs). Thus, FR-S-MCKP can be described as follows:

Initialisation: We set $t=1$ and $c_t = C$, respectively.

Iterative steps:

1. We solve the relaxed knapsack problem described in Eq (6) with residual capacity c_t and remaining time $T-t+1$ (i.e., we replace C and T with c_t and $T-t+1$, respectively). Let $\{\mathbf{x}_\tau^*\}_{\tau=1}^{T+1-t}$ denote the optimal solution, where $\mathbf{x}_\tau^* = \{x_\tau^*(k)\}_{k=0}^K$.
2. We then randomly choose item k with probability $q_t(k) = \frac{\sum_\tau x_\tau^*(k)}{\sum_\tau \sum_k x_\tau^*(k)}$. Let $k^*(c_t,t)$ denote the chosen item with weight $w_{k^*(c_t,t)}$. If $t \geq T$ (we exceed the time limit) or $w_{k^*(c_t,t)} > c_t$ (we exceed the capacity limit) then STOP. Otherwise put item into knapsack.
3. We set $c_{t+1} = c_t - w_{k^*(c_t,t)}$ and $t = t+1$. We repeat Step 1 with the new time and capacity values.

The intuition behind this algorithm is that by choosing an item with probability $q_t(k)$, we can guarantee that the expected value of the chosen item is actually $\frac{1}{T}$ of the optimal solution of the relaxed knapsack. Thus, by summing up over T time steps, the performance of FR-S-MCKP converges to the relaxed optimal solution, which is an upper bound of the optimum of the original S-MCKP. In fact, regarding the performance of FR-S-MCKP, we state the following:

Theorem 1 *Let $K^* = \arg\max_k \frac{\mu_k}{\nu_k}$. Recall that W denotes the upper bound for all the possible weights. The performance loss of FR-S-MCKP, defined of the difference between its expected performance and that of OPT-S-MCKP, is at most*

$$\frac{\mu_{K^*}}{\nu_{K^*}} W \Big(\sqrt{2T \ln(2T)} + \ln T \Big) + V$$

That is, we can guarantee that the expected performance of FR-S-MCKP is at most $\tilde{O}(\sqrt{T})$ less than that of the optimal

solution, where all the logarithmic terms are hidden in the notation $\tilde{O}$. In terms of computational cost, at each time step t, FR-S-MCKP solves the relaxed knapsack problem defined in Eq (6) with $\text{poly}(K)$ cost (where the degree of the polynom is strictly larger than 1), it is clearly faster than OPT-S-MCKP.

Due to space limitations, we omit the proof. However, the main steps of the proof can be sketched as follows. We decompose the total performance loss of FR-S-MCKP into *stepwise losses*. We then provide an upper bound for each stepwise loss with a function f, which is linear to the difference between the current remaining capacity c_t of FR-S-MCKP and the *average capacity* of the optimal solution at the current time step t. We then provide a *martingale* in which the differences between the consecutive elements are upper bounds of the abovementioned capacity differences. By applying the Azuma-Hoeffding inequality on this martingale, we provide an upper bound on how large these differences can be. Finally, by summing up these bounds over T, we obtain the desired performance loss bound.

4.3 Greedy Approximation

We now turn to the discussion of G-S-MCKP, a greedy heuristic for S-MCKP. In this heuristic, at each time step t, we greedily choose the item that, if we are only allowed to solely choose that item in the remaining time steps, will provide the highest total value w.r.t. the remaining capacity. In particular, the item we choose at time step t is the one that satisfies:

$$k^*(c_t,t) = \arg \max_{1\le k\le K} \max_{t\le T'\le T} \Big\{\mu_k(T'+1-t)|c_t \geq (T'+1-t)\sum_{w=1}^{W} wp_k(w)\Big\} \quad (7)$$

That is, at each time step t, we consider all the indices that on average can still fit into the remaining capacity c_t if we have to *consecutively choose that single index* in the remaining part of the process. We then choose the index that provides highest total expected value in the remaining time steps. Hence, the algorithm can be described as follows:

Initialisation: We set $t = 1$ and $c_t = C$, respectively.

Iterative steps:

1. We calculate $k^*(c_t, t)$ according to Eq (7) and choose this item. If $t \geq T$ (we exceed the time limit) or $w_{k^*(c_t,t)} > c_t$ (we exceed the capacity limit) then STOP. Otherwise put item $k^*(c_t, t)$ into the knapsack and GOTO Step 2.
2. We set $c_{t+1} = c_t - w_{k^*(c_t,t)}$ and $t = t + 1$. We repeat Step 1 with the new time and capacity values.

The intuition behind this heuristic is, that in many cases, by solely choosing a fixed item, we can still achieve a good approximation solution for the knapsack problem (see [Kellerer *et al.*, 2004] for more details). Following this idea, this algorithm chooses the best *fixed-item strategy* that can maximise the total value in the remaining time steps, in order to achieve better approximation. This algorithm is computationally very efficient, as evaluating Eq (7) only requires $O(K)$ complexity at each time step. Thus, it is significantly more efficient in terms of computational costs, compared to OPT-S-MCKP and FR-S-MCKP, respectively. On the other hand, unlike FR-S-MCKP, we do not have theoretical guarantees on the performance of G-S-MCKP. However, as we will demonstrate later, that G-S-MCKP can provide good performance in practice.

5 S-MCKP with Unknown Distributions

In this section, we investigate the case when the value-weight distributions are unknown. That is, we do not have the full information about the distribution function of the value-weight pairs of the items. Instead, we can only observe a realisation of these pairs at each time step. In this setting, achieving the optimum is impossible, as it requires the full knowledge of the distribution functions. Given this, the main goal is shifted to the minimisation of the performance loss. As such, we aim to develop algorithms with sub-linear losses. To achieve this goal, we provide a solution that combines the previously proposed algorithms with distribution estimation in order to efficiently fit this case of S-MCKP. In particular, we describe how to maintain an approximate of the unknown value-weight distributions at each time step and how to apply OPT-S-MCKP, FR-S-MCKP, and G-S-MCKP to this sequence of approximated distributions. We then provide rigorous theoretical performance guarantees for these solutions.

5.1 Approximated S-MCKP

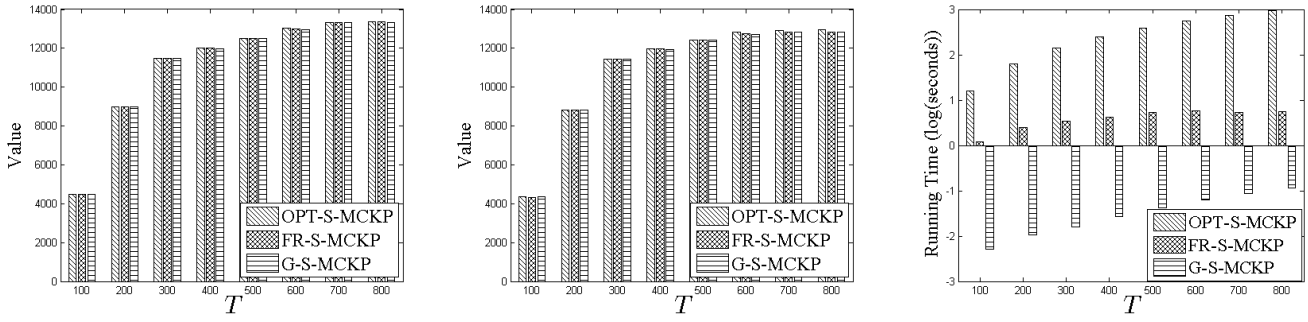
We start with the approximation of the value-weight distribution. As we can observe the K value-weight pairs of each item k at every time step, we can use the empirical distribution of the value-weight distribution, generated by the observed data, to approximate the parameters that are needed for us. In particular, we are interested in the mean of the value and the marginal distribution of the weight of each item, as all the proposed algorithms, (i.e., OPT-S-MCKP, FR-S-MCKP, and G-S-MCKP) use them in order to calculate the next item to choose. Given this, let $\hat{\mu}_{k,t}$ denote the average of the values observed for item k at time step t (not including the value that is revealed at time step t). In addition, let $\hat{p}_{k,t}(w)$ denote the empirical probability. Having these approximations, the algorithms proposed in Section 4 can be applied to the setting of unknown value-weight distribution as follows:

Initialisation: We set $t = 1$ and $c_t = C$. Let ALG denote either OPT-S-MCKP, FR-S-MCKP, and G-S-MCKP, respectively. We randomly choose one item to put in to the knapsack. If we exceed either the time or capacity constraint then STOP, otherwise we update $t = t + 1$ and $c_t = C - w$ where w is the weight of the randomly chosen item. We also update $\hat{\mu}_{k,t}$ and $\hat{p}_{k,t}(w)$ for each k and w, and GOTO next phase[2].

Iterative steps:

1. We run ALG with the current set of $\{\hat{\mu}_{k,t}\}_{k=1}^{K}$ and $\{\hat{p}_{k,t}(w)\}_{k,w}$.
2. If ALG stops (due to exceeding a constraint) then STOP, otherwise $t = t + 1$ and $c_t = c_t - w$ where w is the weight of the currently chosen item, update $\{\hat{\mu}_{k,t}\}_{k=1}^{K}$ and $\{\hat{p}_{k,t}(w)\}_{k,w}$, and GOTO Step 1.

[2]Note that FR-S-MCKP also needs to maintain $\hat{\nu}_{k,t}$, the average weight of each item k. However, this can be easily calculated from $\hat{p}_{k,t}(w)$ by taking the weighted average of the weights.



(a) Known distributions case. (b) *A priori* unknown distributions case. (c) Computational cost

Figure 1: Numerical results for S-MCKP with 50 items.

5.2 Performance Analysis of the Algorithms

We now turn to the performance analysis of the algorithms within the setting of unknown value-weight distributions. Let $K^* = \arg\max_k \frac{\mu_k}{\nu_k}$, and recall that V and W are the upper bounds of all the values and weights, respectively. We state the following theorems:

Theorem 2 *The performance loss of OPT-S-MCKP without the* prior *knowledge of the value-weight distribution, compared to that of a best possible algorithm, is at most*

$$V\Big(2\sqrt{T\ln 2T} + 2K + 1\Big)$$

Theorem 3 *The performance loss of FR-S-MCKP without the* prior *knowledge of the value-weight distribution is at most*

$$\frac{\mu_{K^*}}{\nu_{K^*}} W \ln T + \sqrt{T\ln 2T}\Big(V + (\sqrt{2}+2)\frac{W^2\mu_{K^*}}{2\nu_{K^*}}\Big) + (K+1)2V$$

Again, we omit the proofs due to space limitations. However, they can be sketched as follows. By using the Azuma-Hoeffding and Dvoretzky-Kiefer-Wolfowitz inequalities, we can provide upper bounds on the estimation error of $\hat{\mu}_{k,t}$ and $\hat{p}_{k,t}(w)$, respectively. Using similar techniques from the proof of Theorem 1 with some other technical algebrae, we get an upper bound on the step-wise losses. By summing these bounds, we obtain the desired results.

As we can see, both OPT-S-MCKP and FR-S-MCKP can achieve $\tilde{O}(\sqrt{T})$ performance loss (again, the logarithmic terms are hidden in the notation $\tilde{O}$), which is sub-linear. This guarantees the desired efficiency of the algorithms. On the other hand, similarly to the case of having full knowledge of the value-weight distributions, we do not have any performance guarantees for G-S-MCKP when it is applied to the setting of unknown value-weight distributions. However, as we will demonstrate in the next section, this algorithm can still perform well in practice, compared to its counterparts.

6 Numerical Evaluations

Given the theoretical analysis of the algorithms, we now turn to the empirical evaluation of the proposed algorithms. The main reason behind having this is that we want to investigate whether G-S-MCKP, which does not have theoretical performance guarantees, can provide good performance, compared to that of the other two methods. To do so, we conduct a set of numerical simulations. In these simulations, the values of each item are sampled from the set $\{10, 15, 20, \cdots, 55\}$ and the weights of each item are from $\{5, 10, 15, \cdots, 50\}$. We set the number of items per time step to be 50, and we generate the value-weight distributions as follows. We take the following block of 10 value-weight distributions: $\mathbb{P}_k\{v_t(k) = 5(k+1)\} = p$, $\mathbb{P}_k\{w_t(k) = 5k\} = p$, $\mathbb{P}_k\{v_t(k) = 5(i+1)\} = \frac{1-p}{9}$, $\mathbb{P}_k\{w_t(k) = 5i\} = \frac{1-p}{9}$ $\forall i \in \{1, 2, \cdots, 10\}\backslash\{k\}$, where $k = 1, 2, \ldots 10$. By varying the value of p over the set $\{0.6, 0.5, 0.4, 0.3, 0.25\}$, we get 50 pairs of value-weight distributions. The capacity of the knapsack is 10000, and we run each experiment 100 times[3].

Figure 1 depicts the results of the algorithms. In particular, Figures 1a and 1b show the performance of the algorithms in the cases of known and *a priori* unknown value-weight distributions, while Figure 1c shows the running time of the algorithms, respectively. As we can see, OPT-S-MCKP performs the best, while FR-S-MCKP and G-S-MCKP achieve very close performance to that of OPT-S-MCKP in both cases. In particular, we can observe that FR-S-MCKP is slightly better than G-S-MCKP. This is due to the fact that the fractional relaxation technique typically provides better approximation, compared to the fixed item greedy approach [Kellerer *et al.*, 2004]. As FR-S-MCKP converges to the former and G-S-MCKP converges to the latter, FR-S-MCKP indeed has better performance. We can also observe that the performance of the algorithms in the unknown distributions case is slightly worse than in its known distributions counterpart. This is due to the estimation errors of the distributions. However, as these errors typically converge to 0 in a fast manner, the performance loss we obtain by moving from known distributions to the unknown case is also small. In terms of running time (see Figure 1c), G-S-MCKP achieves the most efficient computational running time that is faster than that of the others by up to an order of magnitude of 2.5 (compared to the running time of FR-S-MCKP) and 4 (compared to OPT-S-MCKP), respectively. In addition, FR-S-MCKP is faster than OPT-S-MCKP by up to a magnitude of 2. Thus, this results demonstrate that both FR-S-MCKP and G-S-MCKP can achieve near optimal performance with significantly less computational cost, com-

[3]We have also run additional experiments with different parameter settings. However, due to page limitations, we ignore the details of those results, as they show similar results on a broad view.

pared to that of OPT-S-MCKP.

7 Conclusions

In this paper we provide a comprehensive analysis of S-MCKP, the stochastic multiple-choice knapsack problem. In particular, we proposed OPT-S-MCKP, an optimal algorithm for the case when the value-weight distributions are known. We also proved that this algorithm can achieve $\tilde{O}(\sqrt{T})$ performance loss when these distributions are not known *a priori*. We also developed two computationally efficient approximation algorithms, FR-S-MCKP and G-S-MCKP. The former can provably achieve $\tilde{O}(\sqrt{T})$ performance loss for both known and unknown distribution scenarios. On the other hand, while G-S-MCKP does not have rigorous theoretical performance guarantees, we demonstrated that it can still achieve good performance (i.e., comparable with the other two methods) with a significantly lower computational cost. As a result, these approximation algorithms can be used in many real-world applications where both low computational cost and high performance are key requirements.

As a possible future work, we aim to provide theoretical performance guarantees for G-S-MCKP, as our conjecture is that it can also achieve $\tilde{O}(\sqrt{T})$ performance guarantee. However, this is not trivial as our current techniques are not suitable for tackling this problem. In particular, it is essential to estimate how bad a decision of G-S-MCKP at each time step is, compared to that of the optimal algorithm.

8 Acknowledgement

We gratefully acknowledge funding from the UK Research Council for project 'ORCHID', grant EP/I011587/1.

References

[Amin *et al.*, 2015] K. Amin, S. Kale, G. Tesauro, and D. Turaga. Budgeted prediction with expert advice. *AAAI*, 2015.

[Babaioff *et al.*, 2007] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 4627 of *Lecture Notes in Computer Science*, pages 16–28. 2007.

[Badanidiyuru *et al.*, 2013] A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. In *IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216, 2013.

[Bagchi *et al.*, 1996] A. Bagchi, N. Bhattacharyya, and N. Chakravarti. Lp relaxation of the two dimensional knapsack problem with box and gub constraints. *European Journal of Operational Research*, 89:609–617, 1996.

[Benoist *et al.*, 2001] T. Benoist, E. Bourreau, Y. Caseau, and B. Rottembourg. Towards stochastic constraint programming: A study of onine multi-choice knapsack with deadlines. In Toby Walsh, editor, *Principles and Practice of Constraint Programming CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 61–76. Springer Berlin Heidelberg, 2001.

[Cesa-Bianchi and Lugosi, 2006] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. 2006.

[Cesa-Bianchi *et al.*, 1997] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R. Schapire, and M.Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[Cohn and Mit, 1998] A. M. Cohn and C. B. Mit. The stochastic knapsack problem with random weights: A heuristic approach to robust transportation planning. In *in Proceedings of the Triennial Symposium on Transportation Analysis*, 1998.

[Ding *et al.*, 2013] W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu. Multi-armed bandit with budget constraint and variable costs. *AAAI*, 2013.

[Ho and Wortman-Vaughan, 2012] C.-J. Ho and J. Wortman-Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, pages 45–51, 2012.

[Kakade *et al.*, 2007] S. M. Kakade, A. T. Kalai, and K. Ligett. Playing games with approximation algorithms. In *STOC*, pages 546–555, 2007.

[Kellerer *et al.*, 2004] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[Lu *et al.*, 1999] L. L. Lu, S. Y. Chiu, and L. A. Cox Jr. Optimal project selection: Stochastic knapsack with finite time horizon. *Journal of the Operational Research Society*, pages 645–650, 1999.

[Lueker, 1995] G. S. Lueker. Average-case analysis of off-line and on-line knapsack problems. In *SODA*, pages 179–188, 1995.

[Tran-Thanh *et al.*, 2010] L. Tran-Thanh, A. Chapman, J. E. Munoz de Cote, A. Rogers, and N. R. Jennings. Epsilon–first policies for budget–limited multi–armed bandits. *AAAI*, pages 1211–1216, 2010.

[Tran-Thanh *et al.*, 2012] L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*, pages 1134–1140, 2012.

[Tran-Thanh *et al.*, 2014] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214(0):89 – 111, 2014.

[Zhou and Naroditskiy, 2008] Y. Zhou and V. Naroditskiy. Algorithm for stochastic multiple-choice knapsack problem and application to keywords bidding. In *WWW*, pages 1175–1176, 2008.