Finding the bandit in a graph: Sequential search-and-stop

Pierre Perrault
SequeL team, INRIA Lille
CMLA, ENS Paris-Saclay
pierre.perrault@inria.fr

Vianney Perchet
CMLA, ENS Paris-Saclay
Criteo Research
vianney.perchet@normalesup.org

Michal Valko
SequeL team
INRIA Lille – Nord Europe
michal.valko@inria.fr

Abstract

We consider the problem where an agent wants to find a hidden object that is randomly located in some vertex of a directed acyclic graph (DAG) according to a fixed but possibly unknown distribution. The agent can only examine vertices whose in-neighbors have already been examined. In this paper, we address a *learning* setting where we allow the agent to stop before having found the object and restart searching on a new independent instance of the same problem. Our goal is to maximize the total number of hidden objects found given a time budget. The agent can thus skip an instance after realizing that it would spend too much time on it. Our contributions are both to the *search theory* and *multi-armed bandits*. If the distribution is known, we provide a quasi-optimal and efficient stationary strategy. If the distribution is unknown, we additionally show how to sequentially approximate it and, at the same time, act near-optimally in order to collect as many hidden objects as possible.

1 Introduction

We study the setting where an object, called hider, is randomly located in one vertex of a directed acyclic graph (DAG), and where an agent wants to find it by sequentially selecting vertices one by one, and examining them at a (possibly random) cost. The agent has a strong constraint: its search must respect *precedence constraints* imposed by the DAG, i.e., a vertex can be examined only if *all* its in-neighbors have already been examined. The goal of the agent is to minimize the expected total search cost incurred before finding the hider. This setting is a type of *single machine scheduling* problem (Lín, 2015), where a set of n jobs $[n] \triangleq \{1, \dots, n\}$ have to be processed on a single machine that can process at most one job at a time. Once a job processing is started, it must continue without interruption until the processing is complete. Each job j has a cost c_j representing its processing time, and a weight w_j representing its importance. In our context, w_j is the probability that j contains the hider. The aim is to find a schedule (i.e., a permutation of jobs) that minimizes the total weighted completion time while respecting precedence constraints[1]. The setting was already shown to be NP-hard (Lawler, 1978; Lenstra and Rinnooy Kan, 1978). On the positive side, several polynomial-time α-approximations exist, depending on the assumption we take on the DAG (see e.g., the recent survey of Prot and Bellenguez-Morineau, 2018). For instance, the case of $\alpha = 2$ can be dealt without any additional assumption. On the other hand, there is an exact $\mathcal{O}(n \log n)$-time algorithm when the partially ordered set (poset) defined by the DAG is a series-parallel order (Lawler, 1978). More generally, when the poset has fractional dimension of at most f, there is a polynomial-time approximation with $\alpha = 2 - 2/f$ (Ambühl et al., 2011). In this work, we assume the DAG is such that an exact polynomial-time algorithm is available. We denote this algorithm as Scheduling. For example, this is true for two-dimensional posets (Ambühl and Mastrolilli, 2009).

The problem is also well known in *search theory* (Stone, 1976; Fokkink et al., 2016), one of the disciplines originating from *operations research*. Since in our case, the search space is a DAG, we fall within the network search setting (Kikuta and Ruckle, 1994; Gal, 2001; Evans and Bishop, 2013). When the DAG is an out-tree, the problem reduces to the *expanding search* problem introduced by Alpern and Lidbetter (2013).

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

[1] The standard scheduling notation (Graham et al., 1979) denotes this setting as $1|prec|\sum w_j C_j$.

The case of unknown distribution of the hider is usually studied within the field of *search games*, i.e., a zero-sum game where the agent picks the search and plays against the hider with search cost as payoff (Alpern and Gal, 2006; Alpern et al., 2013; Hohzaki, 2016). In our work, we deal with an unknown hider distribution by extending the stochastic setting to the sequential case, where at each round t, the agent faces a new, independent instance of the problem. The challenge is the need to learn the distribution through repeated interactions with the environment. Each instance, the agent has to perform a search based on the instances observed during the previous rounds. Furthermore, contrary to the typical search setting, the agent can additionally decide whether it wishes to abandon the search on the current instance and start a new one in the next round, even if the hider was not found. The goal of the agent is to collect as many hiders as possible, using a fixed budget B. This may be particularly useful, when the remaining vertices have large costs and it would not be cost-effective to examine them.

As a result, the hider may not be found in each round and the agent has to make a trade-off between exhaustive searches, which lead to a good estimation (*exploration*) and efficient searches, which leads to a good benefit/cost ratio (*exploitation*). The sequential *exploration-exploitation* trade-off is well studied in multi-armed bandits (Cesa-Bianchi and Lugosi, 2006; Lattimore and Szepesvári, 2019) and has been applied to many fields including mechanism design (Mohri and Munoz, 2014), search advertising (Tran-Thanh et al., 2014) and personalized recommendation (Li et al., 2010). Since several vertices can be visited within each round, our setting can be seen as an instance of stochastic combinatorial semi-bandits (Cesa-Bianchi and Lugosi, 2006, 2012; Gai et al., 2012; Gopalan et al., 2014; Kveton et al., 2015; Combes et al., 2015; Wang and Chen, 2017; Valko, 2016). For this reason, we refer to a vertex $j \in [n]$ as an *arm*. We shall see, however, that this specific semi-bandit problem is challenging. In particular, the agent pays a *non-linear* search cost at each round (with respect to the selected combinatorial action), that additionally depends on the ordering. Moreover, due to the budget constraint, it is also an instance of *budgeted bandits*, also known as *bandits with knapsacks* (Badanidiyuru et al., 2013), in the case of single resource and infinite horizon. We thus evaluate the performance of a learning policy with the (common) notion of *expected (budgeted) regret*. It measures the expected difference, in terms of cumulative reward collected within the budget constraint B, between the learning policy and an *oracle* policy that knows a priori the exact parameters of the problem. Budgeted combinatorial semi-bandits have been already studied by Sankararaman and Slivkins (2017) for several resources, but with a finite horizon. Moreover, their algorithm is efficient only for some specific combinatorial structures (such as matroids). The structure of constraints in sequential search-and-stop is in general more complex.

Motivation There are several motivations behind this setting. One example is the *decision-theoretic troubleshooting* problem of giving a diagnosis for several devices having a malfunctioning component and arriving sequentially to the agent. In many *troubleshooting* applications, we additionally face precedence constraints. These restrictions are imposed to the agent as the ordering of component tests, see e.g., Jensen et al., 2001. Moreover, allowing the agent to *stop* gives a new alternative to the so-called *service call* (Heckerman et al., 1995; Jensen et al., 2001) in order to deal with non-cost-effective vertices: Instead of giving a high cost to an extra action that will automatically find the fault in the device, we give it a zero cost, but do not reward such diagnostic failure. This way, we do not need to estimate any *call-service* cost. This alternative is used, for example, when a new device is sent to the user if the diagnostic fails, with a cost that depends on a disutility for the user: loss of personal data, device reconfiguration, etc. Maximizing the number of hiders found is then analogous to maximizing the number of successful diagnoses.

Another example comes from online advertisement. There are several different actions that might generate a conversion from a user, such as sending one or several emails, displaying one or several ads on a website, buying keywords on search engines, etc. We assume that some precedence constraints are imposed between actions and that a conversion will occur if some sequence of actions is made, for instance, first, display an ad, then send the first email, and finally the second one. As a consequence, the conversion is "hidden", the precedence constraints restrict our access to it, and the agent aims at finding it. However, for some users, finding the correct sequence might be too expensive and it might be more interesting to abandon that specific user to focus on more promising ones.

Related settings Finally, there are several settings related to ours. One of them is stochastic probing (Gupta and Nagarajan, 2013), which differs in the fact that each arm can contain a hider, independently from each other. Another one is the machine learning framework of optimal discovery (Bubeck et al., 2013).

Our contributions One of our main contributions is a stationary *offline policy* (i.e., an algorithm that solves the problem when the distribution is known), for which we prove the approximation guarantees and adapt it in order to fit the online problem. In particular,

we prove that it is quasi-optimal and use SCHEDULING to prove its computational efficiency. Next, we provide a solution when the distribution is unknown to the agent, based on combinatorial upper confidence bounds (CUCB) algorithm from Chen et al. (2016), and UCB-variance (UCB-V) of Audibert et al. (2009). Dealing with variance estimates allows us to sharp the bound on the expected regret, improving the overall dependence on the dimension n compared to the simple use of CUCB. We also propose a new method (that can be of independent interest) to avoid the typical $1/c_{\min}^2$ term in the expected regret bound (Tran-Thanh et al., 2012; Ding et al., 2013; Xia et al., 2016a,b; Watanabe et al., 2017), where $c_{\min}$ is the minimal expected search cost paid over a single round.

2 Background

In this paper, we typeset vectors in bold and indicate components with indices, i.e., $\mathbf{a} = (a_i)_{i\in[n]} \in \mathbb{R}^n$. We formalize in this section the setting we consider. We denote a finite DAG by $\mathcal{G} \triangleq ([n], \mathcal{E})$, where $[n]$ is its set of vertices, or arms, and $\mathcal{E}$ is its set of directed edges. For more generality, we assume arm costs are random and mutually independent. We denote $C_j \in [0,1]$, with expectation $c_j^\star \triangleq \mathbb{E}[C_j] > 0$, the cost of arm j. We thus have $\mathbf{c}^\star = \mathbb{E}[\mathbf{C}] \in (0,1]^n$. We also assume that one specific vertex, called *hider*, is chosen at random, independently from $\mathbf{C}$, accordingly to some fixed categorical (or multivariate Bernoulli) distribution parameterized by vector $\mathbf{w}^\star$ satisfying[2] $\sum_{i=1}^n w_i^\star = 1$ and $w_i^\star \in [0,1]$. Notice that $\mathbf{W} \sim \text{Bernoulli}(\mathbf{w}^\star)$ if, given $i \in [n]$ and with probability $w_i^\star$, $W_i = 1$ and $W_j = 0$ for all $j \neq i$. We also use $\mathcal{D}$ to denote the joint distribution of $(\mathbf{C}, \mathbf{W})$.

Let $\mathbf{e}_i \in \mathbb{R}^n$ denote the i^{th} canonical unit vector. For an (ordered) subset A of $[n]$, we denote by A^c the complementary of A in $[n]$ and $|A|$ its cardinality. The *incidence vector* of A is $\mathbf{e}_A \triangleq \sum_{i\in A} \mathbf{e}_i$. The above definition allows representing a subset of $[n]$ as an element of $\{0,1\}^n$. Let $\mathcal{G}\langle A\rangle$ be the sub-DAG in $\mathcal{G}$ induced by A, i.e., the DAG with A as vertex set, and with (i,j) an arc in $\mathcal{G}\langle A\rangle$ if and only if $(i,j) \in \mathcal{E}$. We call *support* of an ordered arm set $\mathbf{a} = (a_1, \dots, a_k)$ the corresponding non-ordered set. If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we write $\mathbf{x} \geq \mathbf{y}$ (resp., $\mathbf{x} \leq \mathbf{y}$) if $\mathbf{x} - \mathbf{y} \in \mathbb{R}^n_+$ (resp., $\mathbf{y} - \mathbf{x} \in \mathbb{R}^n_+$). We let $\mathbf{a}[j] \triangleq (a_1, a_2, \dots, a_j)$ for $j \leq |\mathbf{a}|$. In addition, we let $\mathbf{a}[\mathbf{W}] \triangleq \mathbf{a}[j]$ if there is j such that $W_{a_j} = 1$, and $\mathbf{a}[\mathbf{W}] \triangleq \mathbf{a}$ otherwise. For two disjoint ordered arm sets $\mathbf{a}$ and $\mathbf{b}$, we let $\mathbf{ab} = (a_1, a_2, \dots, a_{|a|}, b_1, b_2, \dots, b_{|b|})$ be the concatenation of $\mathbf{a}$ and $\mathbf{b}$.

We assume that $\mathcal{G}$ allows a polynomial-time algorithm (w.r.t. n), that takes some parameters $\mathbf{w}, \mathbf{c} \in \mathbb{R}^n_+$, and outputs $\mathbf{s} = \text{SCHEDULING}(\mathbf{w}, \mathbf{c}, \mathcal{G})$ minimizing

$$d(\mathbf{s}; \mathbf{w}, \mathbf{c}) \triangleq \sum_{i=1}^{|\mathbf{s}|} w_{s_i} \mathbf{e}_{\mathbf{s}[i]}^\mathsf{T} \mathbf{c} = \sum_{i=1}^{|\mathbf{s}|} w_{s_i} \sum_{j=1}^{i} c_{s_j}$$

over linear extensions[3] $\mathbf{s} = (s_1, \dots, s_n)$ of the poset defined by $\mathcal{G}$ (that we call $\mathcal{G}$-linear extensions). Notice that $d(\mathbf{s}) \triangleq d(\mathbf{s}; \mathbf{w}^\star, \mathbf{c}^\star)$ represents the expected cost $\mathbb{E}\left[\mathbf{e}_{\mathbf{s}[\mathbf{W}]}^\mathsf{T} \mathbf{C}\right]$ to pay for finding the hider with the $\mathcal{G}$-linear extension $\mathbf{s}$, i.e., by searching arm s_1 first and paying C_{s_1}, then s_2 by paying C_{s_2} in case $W_{s_1} = 0$, and so on until the hider is found, i.e., the last arm i searched is such that $W_i = 1$.

We define a *search* in $\mathcal{G}$ as an ordering $\mathbf{s} = (s_1, \dots, s_k)$ of different arms such that for all $i \in [k]$, predecessors of s_i in $\mathcal{G}$ are included in $\{s_1, \dots, s_{i-1}\}$, i.e., a search is a prefix of a $\mathcal{G}$-linear extension. We denote by $\mathcal{S}_\mathcal{G}$ (or simply $\mathcal{S}$) the set of searches in $\mathcal{G}$. Search supports are called *initial sets*.

2.1 Protocol

Our search setting is *sequential*. We consider an *agent*, also called a *learning algorithm* or a *policy* that knows $\mathcal{G}$ but that does not know $\mathcal{D}$. At each round t, an independent sample $(\mathbf{C}_t, \mathbf{W}_t)$ is drawn from $\mathcal{D}$. The aim of the agent is to search the hider (i.e., the arm i such that $W_{i,t} = 1$) by constructing a *search* on $\mathcal{G}$. Since the hider may be located at some arm that does not belong to the search, it is not necessarily found over each round.

The search to be used by the agent can be chosen based on all its previous observations, i.e., all the costs of explored vertices (and only those) and all the locations where the hider has been found or not. Obviously, the search cannot use the non-observed quantities. For example, the agent may estimate $\mathbf{w}^\star$ and $\mathbf{c}^\star$ in order to choose the search accordingly. Each time an arm j is searched, the feedback $W_{j,t}$ and $C_{j,t}$ is given to the agent. Since several arms can be searched over one round, this problem falls into the family of *stochastic combinatorial semi-bandits*. The agent can keep searching until its budget, B, runs out. B is a positive number and does not need to be known to the agent in advance. The agent wants to maximize the overall number of hiders found under the budget constraint.

The setting described above allows the agent to modify its behavior depending on the feedback it received during the current round. However, by independence

[2] i.e., $\mathbf{w}^\star$ belongs to the simplex of $\mathbb{R}^n$

[3] A linear extension of a poset is a total ordering consistent with the poset, i.e., if a is before b in the poset, then the same has to be true for its linear extension.

assumption between random variables, the only feedback susceptible to modify the search the agent chose at the beginning of a round t is the observation of $W_{i,t} = 1$ for some arm i. Even if nothing prevents the agent from continuing "searching" some arms after having seen such an event, it would not increase the number of hiders found (there is no more hider to find), while this would still decrease the remaining budget, and therefore it would have a pure exploratory purpose. Knowing this, an oracle policy that knows exactly $\mathcal{D}$ thus *selects* a search $\mathbf{s}$ at the beginning of round t, and then *performs* the search that follows $\mathbf{s}$ until either $W_{i,t} = 1$ is observed or $\mathbf{s}$ is exhausted (i.e., no arms are left in $\mathbf{s}$). Therefore, the performed search is in fact $\mathbf{s}[\mathbf{W}_t]$. We thus restrict ourselves to agents that select a search $\mathbf{s}$ at the beginning of each round t and then performs $\mathbf{s}[\mathbf{W}_t]$ over this round. As a consequence, the selected search $\mathbf{s}$ is computed based on observations collected during previous rounds $t-1, t-2, \dots$, denoted $\mathcal{H}_t$, that we refer to as *history*.

Following Stone (1976), we refer to our problem as *sequential search-and-stop*. We now detail the overall objective for this problem: The agent wants to follow a policy π, that selects a search $\mathbf{s}_t$ at round t (this choice can be random as it may depend on the past observations $\mathcal{H}_t$, as well as possible randomness from the algorithm), while maximizing the expected overall reward

$$F_B(\pi) \triangleq \mathbb{E}\left[\sum_{t=1}^{\tau_B - 1} \mathbf{e}_{\mathbf{s}_t[\mathbf{W}_t]}^\intercal \mathbf{W}_t\right] = \mathbb{E}\left[\sum_{t=1}^{\tau_B - 1} \sum_{i \in \mathbf{s}_t[\mathbf{W}_t]} W_{i,t}\right],$$

where τ_B is the random round at which the remaining budget becomes negative: In particular, we have that if $B_t \triangleq B - \sum_{u=1}^{t} \mathbf{e}_{\mathbf{s}_u[\mathbf{W}_u]}^\intercal \mathbf{C}_t$, then $B_{\tau_B - 1} \geq 0$ and $B_{\tau_B} < 0$. We evaluate the performance of a policy using the *expected (budgeted) regret* with respect to $F_B^\star$, the maximum value of F_B (among all possible oracle policies that know $\mathcal{D}$ and B), defined as

$$R_B(\pi) \triangleq F_B^\star - F_B(\pi).$$

Example 1. *One may wonder if there exist cases where it is interesting for the agent to stop the search earlier. Consider for instance the simplest non-trivial case with two arms and no precedence constraint. The costs are deterministically chosen to be ε and 1 and the location of the hider is chosen uniformly at random. An optimal search will always first sample the arm with $\varepsilon < 1$ cost. If it also samples the other one, then the hider will be found with an expected cost of $\varepsilon + 1/2$. However, if the agent always stops the search after the first arm, and reinitializes on a new instance by doing the same, the overall cost to find one hider is*

$$\sum_{t=1}^{\infty} \left(\frac{1}{2}\right)^t t\varepsilon = 2\varepsilon < \varepsilon + \frac{1}{2}, \qquad \text{for } \varepsilon < \frac{1}{2}.$$

Therefore, stopping searches, even if the location of the hider is known, may be better than always trying to find it.

3 Offline oracle

In this section, we provide an algorithm for sequential search-and-stop when parameters $\mathbf{w}^\star$ and $\mathbf{c}^\star$ are given to the agent. We show that a simple stationary policy (i.e., the same search $\mathbf{s}^\star$ is selected at each round) can obtain almost the same expected overall reward as $F_B^\star$. We will denote by Oracle an algorithm that takes $\mathbf{w}^\star$, $\mathbf{c}^\star$, and $\mathcal{G}$ as input and outputs $\mathbf{s}^\star$. This *offline* oracle will eventually be used by the agent for the *online* problem, i.e., when parameters are unknown. Indeed, at round t, the agent can approximate $\mathbf{s}^\star$ by the output $\mathbf{s}_t$ of Oracle$(\mathbf{w}_t, \mathbf{c}_t, \mathcal{G})$, where $\mathbf{w}_t, \mathbf{c}_t$ can be any guesses/estimates of the true parameters. Importantly, depending on the estimation followed by the agent, $\mathbf{w}_t$ may not stay in the simplex anymore. We will thus build Oracle such that an "acceptable" output is given for any input $(\mathbf{w}, \mathbf{c}) \in \left(\mathbb{R}_+^n\right)^2$.

3.1 Objective design

A standard paradigm for designing a stationary approximation of the offline problem in budgeted multi-armed bandits is the following: $\mathbf{s}^\star$ has to minimize the ratio between the expected cost paid and the expected reward gain, over a single round, selecting $\mathbf{s}^\star$. We thus define, for $\mathbf{s} \in \mathcal{S}$,

$$\begin{aligned} J(\mathbf{s}) &\triangleq \mathbb{E}\left[\mathbf{e}_{\mathbf{s}[\mathbf{W}]}^\intercal \mathbf{C}\right] \mathbb{E}\left[\mathbf{e}_{\mathbf{s}[\mathbf{W}]}^\intercal \mathbf{W}\right]^{-1} \\ &= \frac{d(\mathbf{s}) + (1 - \mathbf{e}_{\mathbf{s}}^\intercal \mathbf{w}^\star)\mathbf{e}_{\mathbf{s}}^\intercal \mathbf{c}^\star}{\mathbf{e}_{\mathbf{s}}^\intercal \mathbf{w}^\star} \\ &= \sum_{i=1}^{|\mathbf{s}|} \frac{c_{s_i}^\star \left(1 - \mathbf{e}_{\mathbf{s}[i-1]}^\intercal \mathbf{w}^\star\right)}{\mathbf{e}_{\mathbf{s}}^\intercal \mathbf{w}^\star}. \end{aligned}$$

Notice that we allow J to be equal to $+\infty$ (when $\mathbf{e}_{\mathbf{s}}^\intercal \mathbf{w}^\star = 0$). We use the convention $J(\emptyset) = +\infty$, because there is no interest in choosing an empty search for a round. We define the optimal values of J on $\mathcal{S}$ as

$$J^\star \triangleq \min_{\mathbf{s} \in \mathcal{S}} J(\mathbf{s}), \quad \mathcal{S}^\star \triangleq \operatorname{argmin}_{\mathbf{s} \in \mathcal{S}} J(\mathbf{s}).$$

We now provide guarantees for this stationary policy.

Proposition 1. *If $\pi^\star$ is the offline policy selecting $\mathbf{s}^\star \in \mathcal{S}^\star$ at each round t, then*

$$\frac{B - n}{J^\star} \leq F_B(\pi^\star) \leq F_B^\star \leq \frac{B + n}{J^\star}.$$

A proof is given in Appendix B and follows the one provided for Lemma 1 of Xia et al. (2016b). Intuitively,

Proposition 1 states that the optimal overall expected reward that can be gained (i.e., the maximum expected number of hiders found) is approximately $B/J^\star$ (we assume that $B \gg n$). This is quite intuitive, since this quantity is actually the ratio between the overall budget and the minimum expected cost paid to find a *single* hider. Indeed, one can consider the related problem of minimizing the overall expected cost paid, over several rounds, to find a single hider. It can be expressed as an infinite-time horizon Markov decision process (MDP) with action space $\mathcal{S}$ and two states: whether the hider is found (which is the terminal state) or not. The goal is to choose a strategy $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_t, \dots$, minimizing

$$\mathcal{J}(\mathbf{s}_1, \mathbf{s}_2, \dots) \triangleq \mathbb{E}\left[\sum_{t=1}^{\tau} \mathbf{e}_{\mathbf{s}_t[\mathbf{W}_t]}^\intercal \mathbf{C}_t\right]$$
$$= \sum_{t=1}^{\infty}\left(\mathbf{e}_{\mathbf{s}_t}^\intercal \mathbf{w}^\star \left(\sum_{u=1}^{t-1} \mathbf{e}_{\mathbf{s}_u}^\intercal\right)\mathbf{c}^\star + d(\mathbf{s}_t)\right)\prod_{u=1}^{t-1}\left(1 - \mathbf{e}_{\mathbf{s}_u}^\intercal \mathbf{w}^\star\right),$$

where the stopping time τ is the first round at which the hider is found. The Bellman equation is

$$\mathcal{J}(\mathbf{s}_1, \mathbf{s}_2, \dots) = d(\mathbf{s}_1) + \left(1 - \mathbf{e}_{\mathbf{s}_1}^\intercal \mathbf{w}^\star\right)\left(\mathbf{e}_{\mathbf{s}_1}^\intercal \mathbf{c}^\star + \mathcal{J}(\mathbf{s}_2, \dots)\right),$$

from which we deduce there exists an optimal *stationary* strategy (Sutton and Barto, 1998) such that $\mathbf{s}_t = \mathbf{s}$ for all $t \in \mathbb{N}^\star$. Therefore, we can minimize $\mathcal{J}(\mathbf{s}, \mathbf{s}, \dots) = J(\mathbf{s})$ that gives the optimal value of $J^\star$.

As we already mentioned, Oracle aims at taking inputs $(\mathbf{w}, \mathbf{c}) \in \left(\mathbb{R}_+^n\right)^2$. The first straightforward way to do is to consider

$$J(\mathbf{s}; \mathbf{w}, \mathbf{c}) \triangleq \sum_{i=1}^{|\mathbf{s}|} \frac{c_{s_i}\left(1 - \mathbf{e}_{\mathbf{s}[i-1]}^\intercal \mathbf{w}\right)}{\mathbf{e}_{\mathbf{s}}^\intercal \mathbf{w}}.$$

However, notice that with the definition above, $J(\,\cdot\,; \mathbf{w}, \mathbf{c})$ could output negative values (if $\mathbf{e}_{[n]}^\intercal \mathbf{w} > 1$), which is not desired, because the agent would then be enticed to search arms with a high cost. We thus need to design a non-negative extension of J to $(\mathbf{w}, \mathbf{c}) \in \left(\mathbb{R}_+^n\right)^2$. One way is to replace $\left(1 - \mathbf{e}_{\mathbf{s}[i-1]}^\intercal \mathbf{w}\right)$ by $\mathbf{e}_{(\mathbf{s}[i-1])^c}^\intercal \mathbf{w}$, another is to consider $J(\mathbf{s}; \mathbf{w}, \mathbf{c})^+$, where $x^+ \triangleq \max\{0, x\}$. There is a significant advantage of considering the second way, even if it is less natural than the first one, which is that for $(\mathbf{w}, \mathbf{c}) \in \left(\mathbb{R}_+^n\right)^2$,

$$J(\mathbf{s}; \mathbf{w}, \mathbf{c})^+ \le J(\mathbf{s}; \mathbf{w}^\star, \mathbf{c}^\star) = J(\mathbf{s}),$$

if $\mathbf{w} \ge \mathbf{w}^\star$ and $\mathbf{c} \le \mathbf{c}^\star$. This property[4] is known to be useful for analysis of many stochastic combinatorial semi-bandit algorithms (see e.g., Chen et al., 2016). Thus, we choose for Oracle the minimization of the surrogate $J(\,\cdot\,; \mathbf{w}, \mathbf{c})^+$.

[4]Notice this is not exactly a monotonicity property, because we compare to a single point $(\mathbf{w}^\star, \mathbf{c}^\star)$.

3.2 Algorithm and guarantees

We now provide Oracle in Algorithm 1 and claim in Theorem 1 that it minimizes $J(\,\cdot\,; \mathbf{w}, \mathbf{c})^+$ over $\mathcal{S}$. Notice that Oracle needs to call the polynomial-time algorithm Scheduling$(\mathbf{w}, \mathbf{c}, \mathcal{G})$, that minimizes the objective function $d(\mathbf{s}; \mathbf{w}, \mathbf{c})$ over $\mathcal{G}$-linear extensions $\mathbf{s}$. Then, Algorithm 1 only computes the maximum value index of a list of size n that takes linear time. To give an intuition, $\mathbf{s}^\star$ follows the ordering given by Scheduling$(\mathbf{w}, \mathbf{c}, \mathcal{G})$, and stops at some point when it becomes more interesting to start a fresh new instance.

Algorithm 1 Oracle

Input: $\mathbf{w}, \mathbf{c}$ and $\mathcal{G}$.
$\mathbf{s} \triangleq$ Scheduling$(\mathbf{w}, \mathbf{c}, \mathcal{G})$.
$i^\star \triangleq \operatorname{argmin}_{i \in [n]} J(\mathbf{s}[i]; \mathbf{w}, \mathbf{c})^+$ (ties may be broken arbitrarily).
Output: the search $\mathbf{s}^\star \triangleq \mathbf{s}[i^\star]$.

Theorem 1. *For every $(\mathbf{w}, \mathbf{c}) \in \left(\mathbb{R}_+^n\right)^2$, Algorithm 1 outputs a search minimizing $J(\,\cdot\,; \mathbf{w}, \mathbf{c})^+$ over $\mathcal{S}$.*

We provide a proof of Theorem 1 in Appendix A. It mixes known concepts of scheduling theory, such as Sidney decomposition (Sidney, 1975), with our new results for our objective function.

4 Online search-and-stop

In this section, we consider an additional challenge where the distribution $\mathcal{D}$ is unknown and the agent must deal with it, while minimizing $R_B(\pi)$ over sampling policies π, where B is a fixed budget. Recall that a policy π selects a search $\mathbf{s}_t$ at the beginning of round t, using previous observations $\mathcal{H}_t$, and then performs the search $\mathbf{s}_t[\mathbf{W}_t]$ over the round. We treat the setting as a variant of stochastic combinatorial semi-bandits (Gai et al., 2012). The feedback received by an agent at round t is random, because it depends on $\mathbf{s}_t$. However, unlike in similar settings, it also depends on $\mathbf{W}_t$, and thus it is not measurable w.r.t. $\mathcal{H}_t$. More precisely, $(W_{i,t}, C_{i,t})$ is observed only for arms $i \in \mathbf{s}_t[\mathbf{W}_t]$. Notice that since $\mathbf{W}_t$ is a one-hot vector, the agent can always deduce the value of $W_{i,t}$ for all $i \in \mathbf{s}_t$. As a consequence, we will maintain two types of counters for all arms $i \in [n]$ and all $t \ge 1$,

$$N_{\mathbf{w},i,t-1} \triangleq \sum_{u=1}^{t-1} \mathbb{I}\{i \in \mathbf{s}_u\},$$
$$N_{\mathbf{c},i,t-1} \triangleq \sum_{u=1}^{t-1} \mathbb{I}\{i \in \mathbf{s}_u[\mathbf{W}_u]\}. \tag{1}$$

We define the corresponding empirical averages[5] as

$$\begin{aligned}\overline{w}_{i,t-1} &\triangleq \frac{\sum_{u=1}^{t-1}\mathbb{I}\{i\in\mathbf{s}_u\}W_{i,u}}{N_{\mathbf{w},i,t-1}},\\ \overline{c}_{i,t-1} &\triangleq \frac{\sum_{u=1}^{t-1}\mathbb{I}\{i\in\mathbf{s}_u[\mathbf{W}_u]\}C_{i,u}}{N_{\mathbf{c},i,t-1}}.\end{aligned} \tag{2}$$

We propose an approach similar to UCB-V of Audibert et al. (2009), based on CUCB of Chen et al. (2016), called CUCB-V, that uses a variance estimation of $\mathbf{w}^\star$ in addition to the empirical average. Notice that the variance of W_i for an arm i is $\sigma_i^2 \triangleq w_i^\star(1-w_i^\star)$. Furthermore, since W_i is binary, the empirical variance of W_i after t rounds is $\overline{w}_{i,t}(1-\overline{w}_{i,t})$. For every round t and every edge $i\in[n]$, with the previously defined empirical averages, we use the confidence bounds[6] as

$$c_{i,t} \triangleq \left(\overline{c}_{i,t-1} - \sqrt{\frac{0.5\zeta\log t}{N_{\mathbf{c},i,t-1}}}\right)^+,$$

$$w_{i,t} \triangleq \min\left\{\overline{w}_{i,t-1} + \sqrt{\frac{2\zeta\overline{w}_{i,t-1}(1-\overline{w}_{i,t-1})\log t}{N_{\mathbf{w},i,t-1}}} + \frac{3\zeta\log t}{N_{\mathbf{w},i,t-1}}, 1\right\},$$

where we choose the exploration factor to be $\zeta \triangleq 1.2$. Notice that we could take any $\zeta > 1$ as shown by Audibert et al. (2009). We provide the policy $\pi_{\text{CUCB-V}}$ that we consider in Algorithm 2.

Algorithm 2 Combinatorial upper confidence bounds with variance estimates (CUCB-V) for sequential search-and-stop

Input: $\mathcal{G}$.
for $t = 1..\infty$ **do**
  select $\mathbf{s}_t$ given by $\textsc{Oracle}(\mathbf{w}_t, \mathbf{c}_t, \mathcal{G})$.
  perform $\mathbf{s}_t[\mathbf{W}_t]$.
  collect feedback and update counters and empirical averages according to (1) and (2).
end for

4.1 Analysis

Notice that since an arm $i\in\mathbf{s}_t$ is pulled (and thus $C_{i,t}$ is revealed to the agent) with probability $1-\mathbf{e}_{\mathbf{s}[i-1]}^\intercal\mathbf{w}^\star$ over round t, we fall into the setting of *probabilistically triggered arms* w.r.t. costs, described by Chen et al. (2016) and Wang and Chen (2017). Thus we could rely on these prior results. However, the main difficulty in our setting is that we also need to deal with probabilities $W_{i,t}$, that the agent actually observes for every arm i in $\mathbf{s}_t$, either because it actually pulls arm i, or because it deduces the value from other pulls of round t. In particular, if we follow the analysis of Chen et al. (2016) and Wang and Chen (2017), the double sum in the definition of J leads to expected regret bound that is quite large. Indeed, assuming that all costs are deterministically equal to 1, if we suffer an error of δ when approximating each $w_i^\star$, then the global error can be as large as $\sum_{i=1}^n\sum_{j=1}^{i-1}\delta = \mathcal{O}(n^2\delta)$, contrary to just $\mathcal{O}(n\delta)$ for the approximation error w.r.t. costs, that is more common in combinatorial semi-bandits. Thus, we rather combine their work with the variance estimates of $w_i^\star$. Often, this does not provide a significant improvement over UCB in terms of expected regret (otherwise we could do the same for the costs), but since in our case, the variance is of order $1/n$, the gain is non-negligible.[7] We let $c_{\min} > 0$ be any deterministic lower bound on the set $\left\{\mathbf{e}_{\mathbf{s}_u[\mathbf{W}_u]}^\intercal\mathbf{c}^\star, u\geq 1\right\}$. Furthermore, we let

$$T_B \triangleq \lceil 2B/c_{\min}\rceil$$

and for any search $\mathbf{s}$, we define the *gap* of $\mathbf{s}$ as

$$\begin{aligned}\Delta(\mathbf{s}) &\triangleq \mathbf{e}_{\mathbf{s}}^\intercal\mathbf{w}^\star\left(\frac{J(\mathbf{s})}{J^\star}-1\right)\\ &= \frac{1}{J^\star}\sum_{i=1}^{|\mathbf{s}|}c_{s_i}^\star\left(1-\sum_{j=1}^{i-1}w_{s_j}^\star\right) - \sum_{i=1}^{|\mathbf{s}|}w_{s_i}^\star \geq 0,\end{aligned}$$

that represents the *local regret* of selecting a suboptimal search $\mathbf{s}$ at some round. In addition, for each arm $i\in[n]$, we define

$$\Delta_{i,\min} \triangleq \inf_{\mathbf{s}\notin\mathcal{S}^\star:\ i\in\mathbf{s}}\Delta(\mathbf{s}) > 0.$$

We provide bounds for the expected regret of $\pi_{\text{CUCB-V}}$ in Theorem 2. The first bound is $\mathcal{D}$-dependent, and is characterized by $c_{\min}$, $J^\star$, and σ_i^2 , $\Delta_{i,\min}$, $i\in[n]$. Its main term scales logarithmically w.r.t. B. The second bound is true for any sequential search-and-stop problem instance having a fixed value of $c_{\min} > 0$ and $J^\star > 0$.

Theorem 2. *The expected regret of* CUCB-V *satisfies*

$$R_B(\pi_{\text{CUCB-V}}) = \mathcal{O}\left(n\log T_B\sum_{i\in[n]}\frac{1+(J^\star+n)^2\sigma_i^2}{{J^\star}^2\Delta_{i,\min}} + \frac{(J^\star+n)}{J^\star}\log\left(\frac{n}{J^\star\Delta_{i,\min}}\right)\right).$$

In addition,

$$\sup R_B(\pi_{\text{CUCB-V}}) = \mathcal{O}\left(\sqrt{n}\left(1+\frac{n}{J^\star}\right)\sqrt{T_B\log T_B}\right),$$

[5] With the convention $0/0 = 0$.
[6] With the convention $x/0 = +\infty,\ \forall x\geq 0$.
[7] The error δ is thus scaled by the standard deviation, of order $1/\sqrt{n}$, giving a global error of $\mathcal{O}(n^{1.5}\delta)$. We therefore recover the factor $n^{1.5}$ given in Theorem 2.

where the sup *is taken over all possible sequential search-and-stop problems with fixed* $c_{\min}$ *and* $J^\star$.

The proof is in Appendix C. Recall the main challenge comes from the estimation of $\mathbf{w}^\star$ and not from $\mathbf{c}^\star$. Our analysis uses *triggering probability groups* and the *reverse amortization trick* of Wang and Chen (2017) for dealing with costs. However, for hider probabilities, only the second trick is necessary.[8] We use it not only to deal with the slowly concentrating confidence term for the estimates of each arm i, but also to completely amortize the additional fast-rate confidence term due to variance estimation coming from the use of Bernstein's inequality. However, the analysis of Wang and Chen (2017) only considers a deterministic horizon. In our case, we need to deal with a *random-time* horizon. For that, notice that their regret upper bounds that hold in expectation are obtained by splitting the expectation into two parts. The first part is filtered with a high-probability event on which the regret grows as the logarithm of the random horizon and the second one is filtered with a low-probability event, on which we bound the regret by a constant. Since the log function is concave, we can upper bound the expected regret by a term growing as the logarithm of the expectation of the random horizon, with Jensen's inequality. Finally, we upper bound the expectation of the random horizon to get the rate of $\log T_B$.

4.2 Tightness of our regret bounds

Since we succeeded in reducing the dependence on n in the expected regret with confidence bounds based on variance estimates, we can now ask whether this dependence in Theorem 2 is tight. We stress that our solution to sequential search-and-stop is *computationally efficient*. In particular, *both* the offline oracle optimization and the computation of the optimistic search $\mathbf{s}_t$ in the online part *are tractable*.

Whenever rewards are not arbitrary correlated (as is the case in our setting), we can potentially exploit these correlations in order to reduce the regret's dependence on n even further. This could be done by choosing a tighter confidence region such as a confidence ellipsoid (Degenne and Perchet, 2016), or a KL-confidence ball (Combes et al., 2015) instead of coordinate-wise confidence intervals. Unfortunately, these do not lead to computationally efficient algorithms. Notice that given an *infinite* computational power, our dependence on n is not tight. In particular, there is an extra $\sqrt{n}$ factor in our gap-free bound (see Theorem 3). It is an open question whether a better *efficient* policy exists.

[8] When we select search $\mathbf{s}$, all feedback W_i, $i \in \mathbf{s}$ is received with probability 1, so *triggering probability groups* are not useful.

To show that we are only a $\sqrt{n}$ factor away, in the following theorem we provide a class of sequential search-and-stop problems (parameterized by n and B) on which the regret bound provided in Theorem 2 is tight up to a $\sqrt{n}$ factor (and a logarithmic one).

Theorem 3. *For simplicity, let us assume that n is even and that B is a multiple of n. For any optimal online policy π, there is a sequential search-and-stop problem with n arms and budget B such that*

$$-4 + \frac{1}{28}\sqrt{\frac{B}{n}} \le R_B(\pi) = \mathcal{O}\left(\sqrt{B \log\left(\frac{B}{n}\right)}\right).$$

For the proof, we consider a DAG composed of two disjoint paths (Figure 1), with all costs deterministically set to 1 and with the hider located either at $a_{n/2}$ or $b_{n/2}$. This information is given to the agent. We then reduced this setting to a two-arm bandit over at least B/n rounds. The complete proof is in Appendix D.

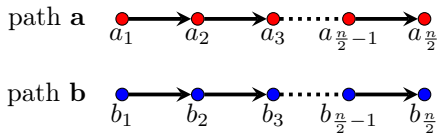

Figure 1: The DAG considered in Theorem 3.

Notice that bounds provided in Theorem 3 *decrease* with n. This is because, in the sequential search-and-stop problem, the increasing dependence on n is counterbalanced by the fact that the number of rounds is of order B/n, and that $J^\star$ is of order n.

5 Experiment

In this section, we present an experiment for sequential search-and-stop. We compare our CUCB-V with three other online algorithms, which are same as CUCB-V except for the estimator $\mathbf{w}_t$ to be plugged in Oracle. We give corresponding definitions of $\mathbf{w}_t$ in Table 1, where we take $\zeta \triangleq 1.2$, and where

$$\mathrm{kl}(p, q) \triangleq p \log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$$

is the Kullback-Leibler divergence between two Bernoulli distributions of parameters $p, q \in [0, 1]$ respectively. We run simulations for all the algorithms with $n = 100$ and without precedence constraints, i.e., when the DAG is an edgeless graph. Notice that in this case, a search can be any ordered subset of arms (thus, the set of possible searches is of cardinality $\sum_{k=0}^{n} n!/k! \le en!$). This restriction does not remove complexity from the online problem, but rather from the offline one, so even

Table 1: Comparison algorithms in the experiment.

Algorithm	Definition of $w_{i,t}$
CUCB	$\min\left\{\overline{w}_{i,t-1} + \sqrt{\frac{0.5\zeta \log t}{N_{\mathbf{w},i,t-1}}}, 1\right\}$
CUCB-KL	The unique solution x to $N_{\mathbf{w},i,t-1}\mathrm{kl}(\overline{w}_{i,t-1}, x) = \zeta \log t$ such that $x \in [\overline{w}_{i,t-1}, 1]$
Thompson Sampling	An independent sample from $\mathrm{Beta}(\alpha, N_{\mathbf{w},i,t-1} - \alpha)$, where $\alpha = N_{\mathbf{w},i,t-1}\overline{w}_{i,t-1}$

in that case, the online problem is challenging. We take parameter $\mathbf{w}^\star$ defined as

$$\begin{cases} w_i^\star = \dfrac{1}{2^i} & \text{for } i \in [m-1] \\ w_m^\star = \left(\dfrac{1}{2} + \varepsilon\right) w_{m-1}^\star \\ w_i^\star = \left(\dfrac{1}{2} - \varepsilon\right) \dfrac{w_{m-1}^\star}{n-m} & \text{for } i \in \{m+1, \dots, n\}, \end{cases}$$

where we chose $m \triangleq 40$. For $\varepsilon \in (0, 1/2)$, one can see that $\mathcal{S}^\star = \{[m]\}$. Intuitively, $w_i^\star$ models the proportion of users answering i to some fixed request:[9] When $\varepsilon = 0$, half of the population answers 1, a quarter answers 2, ..., until m, and remaining users answer uniformly on remaining arms $\{m+1, \dots, n\}$. We chose $\varepsilon = 0.1$, $c_i^\star = 1/2$ for all $i \in [n]$ and take $C_i \sim \mathrm{Bernoulli}(c_i^\star)$. In Figure 2, for each algorithm considered, we plot the quantity

$$\frac{B}{J^\star} - \sum_{t=1}^{\tau_B - 1} \mathbf{e}_{\mathbf{s}_t[\mathbf{W}_t]}^\intercal \mathbf{W}_t,$$

with respect to budget B, averaged over 100 simulations. As shown in Proposition 1, the curves obtained this way provide good approximations to the true regret curves. We notice that CUCB-KL, CUCB-V, and ThompsonSampling are significantly better than CUCB, since the latter explores too much. In addition, the regret curves of CUCB-KL, CUCB-V and ThompsonSampling are quite similar. In particular, their asymptotic slopes seem equal, which hints that regret rates are comparable on this instance.

6 Conclusion and future work

We presented sequential search-and-stop problem and provided a stationary offline solution. We gave theoretical guarantees on its optimality and proved that it is computationally efficient. We also considered the learning extension of the problem where the distribution of the hider and the cost are not known. We provided CUCB-V, an upper-confidence bound approach, tailored to our case and gave expected regret guarantees with respect to the optimal policy.

[9] For recommender systems or search engines, $w_i^\star$ can thus be seen as the probability that an user aims to find i when entering the request.

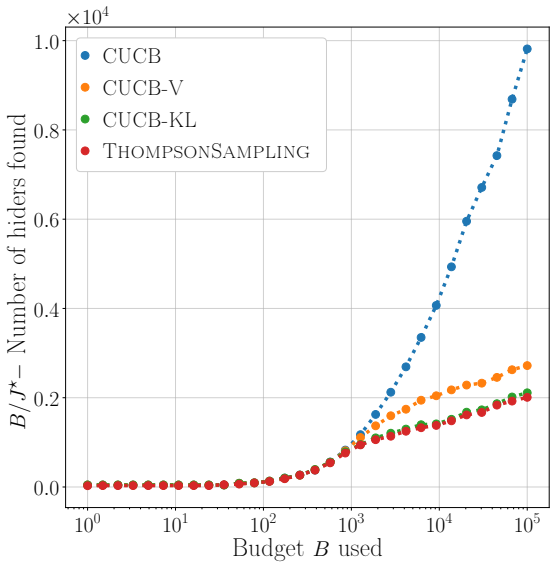

Figure 2: Cumulative regret for sequential search-and-stop, with B up to 10^5, averaged over 100 independent simulations.

We now discuss several possible extensions of our work. We could consider several hiders rather than just one. Another would be to explore the Thomson sampling (Chapelle and Li, 2011; Agrawal and Goyal, 2012; Komiyama et al., 2015; Wang and Chen, 2018) further in the learning case by considering a Dirichlet prior on the *whole* arm set. The Dirichlet seems appropriate because a sample $\mathbf{w}$ from this prior is in the simplex. The main drawback however is the difficulty of *efficiently* updating such prior to get the posterior, because in the case when the hider is not found, the one-hot vector is not received entirely.

Acknowledgements Vianney Perchet has benefited from the support of the ANR (grant n.ANR-13-JS01-0004-01), of the FMJH Program Gaspard Monge in optimization and operations research (supported in part by EDF), from the Labex LMH and from the CNRS through the PEPS program. The research presented was also supported by European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-european project Allocate, and French National Research Agency projects ExTra-Learn (grant n.ANR-14-CE24-0010-01) and BoB (grant n.ANR-16-CE23-0003), FMJH Program PGMO with the support to this program from Criteo.

References

Agrawal, S. and Goyal, N. (2012). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*.

Alpern, S., Fokkink, R., Gasieniec, L., Lindelauf, R., and Subrahmanian, V. S. (2013). *Search theory*. Springer.

Alpern, S. and Gal, S. (2006). *The theory of search games and rendezvous*. Springer.

Alpern, S. and Lidbetter, T. (2013). Mining coal or finding terrorists: The expanding search paradigm. *Operations Research*, 61(2):265–279.

Ambühl, C. and Mastrolilli, M. (2009). Single machine precedence constrained scheduling is a vertex cover problem. *Algorithmica*, 53(4):488–503.

Ambühl, C., Mastrolilli, M., Mutsanas, N., and Svensson, O. (2011). On the approximability of single-machine scheduling with precedence constraints. *Mathematics of Operations Research*, 36(4):653–669.

Audibert, J. Y., Munos, R., and Szepesvári, C. (2009). Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 207–216.

Bubeck, S., Ernst, D., and Garivier, A. (2013). Optimal discovery with probabilistic expert advice: finite time analysis and macroscopic optimality. *Journal of Machine Learning Research*, 14:601–623.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Cesa-Bianchi, N. and Lugosi, G. (2012). Combinatorial bandits. In *Journal of Computer and System Sciences*, volume 78, pages 1404–1422.

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Neural Information Processing Systems*.

Chen, W., Wang, Y., and Yuan, Y. (2016). Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17.

Combes, R., Talebi, M. S., Proutière, A., and Lelarge, M. (2015). Combinatorial bandits revisited. In *Neural Information Processing Systems*.

Degenne, R. and Perchet, V. (2016). Combinatorial semi-bandit with known covariance. *CoRR*, abs/1612.01859.

Ding, W., Qin, T., Zhang, X.-d., and Liu, T.-y. (2013). Multi-Armed Bandit with Budget Constraint and Variable Costs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Evans, T. P. O. and Bishop, S. R. (2013). Static search games played over graphs and general metric spaces. *European Journal of Operational Research*, 231(3):667–689.

Flajolet, A. and Jaillet, P. (2015). Logarithmic regret bounds for Bandits with Knapsacks. *arXiv preprint*.

Fokkink, R., Lidbetter, T., and Végh, L. A. (2016). On submodular search and machine scheduling. *arXiv preprint*.

Gai, Y., Krishnamachari, B., and Jain, R. (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *Transactions on Networking*, 20(5):1466–1478.

Gal, S. (2001). On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory*, 29(4):533–542.

Gopalan, A., Mannor, S., and Mansour, Y. (2014). Thompson sampling for complex online problems. In *International Conference on Machine Learning*.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. H. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5(C):287–326.

Gupta, A. and Nagarajan, V. (2013). A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization*, volume 7801, pages 205–216.

Heckerman, D., Breese, J. S., and Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30.

Hohzaki, R. (2016). Search games: Literature and survey. *Journal of the Operations Research Society of Japan*, 59(1):1–34.

Jensen, F. V., Kjaerulff, U., Kristiansen, B., Langseth, H., Skaanning, C., Vomlel, J., and Vomlelova, M. (2001). The SACSO methodology for troubleshooting complex systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(4):321–333.

Kikuta, K. and Ruckle, W. H. (1994). Initial point search on weighted trees. *Naval Research Logistics*, 41(6):821–831.

Komiyama, J., Honda, J., and Nakagawa, H. (2015). Optimal regret analysis of Thompson sampling in stochastic multi-armed bandit problem with multiple plays. *International Conference on Machine Learning*.

Kveton, B., Wen, Z., Ashkan, A., and Szepesvari, C. (2015). Tight regret bounds for stochastic combinatorial semi-bandits. In *International Conference on Artificial Intelligence and Statistics*.

Lattimore, T. and Szepesvári, C. (2019). *Bandit algorithms*.

Lawler, E. L. (1978). Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2(C):75–90.

Lenstra, J. K. and Rinnooy Kan, A. H. G. (1978). Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. *International World Wide Web Conference*.

Lín, V. (2015). Scheduling results applicable to decision-theoretic troubleshooting. *International Journal of Approximate Reasoning*, 56(PA):87–107.

Mohri, M. and Munoz, A. (2014). Optimal regret minimization in posted-price auctions with strategic buyers. In *Neural Information Processing Systems*.

Prot, D. and Bellenguez-Morineau, O. (2018). A survey on how the structure of precedence constraints may change the complexity class of scheduling problems. *Journal of Scheduling*, 21(1):3–16.

Sankararaman, K. A. and Slivkins, A. (2017). Combinatorial Semi-Bandits with Knapsacks.

Sidney, J. B. (1975). Decomposition Algorithms for Single-Machine Sequencing with Precedence Relations and Deferral Costs. *Operations Research*, 23(2):283–298.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics*, 3(1-2):59–66.

Stone, L. D. (1976). *Theory of optimal search*. Elsevier.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tran-Thanh, L., Chapman, A. C., Rogers, A., and Jennings, N. R. (2012). Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits.

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

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, New York, NY.

Valko, M. (2016). *Bandits on graphs and structures*. habilitation, École normale supérieure de Cachan.

Wang, Q. and Chen, W. (2017). Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Neural Information Processing Systems*.

Wang, S. and Chen, W. (2018). Thompson Sampling for Combinatorial Semi-Bandits.

Watanabe, R., Komiyama, J., Nakamura, A., and Kudo, M. (2017). Kl-ucb-based policy for budgeted multi-armed bandits with stochastic action costs. E100.A:2470–2486.

Xia, Y., Ding, W., Zhang, X.-D., Yu, N., and Qin, T. (2016a). Budgeted bandit problems with continuous random costs. In *Asian Conference on Machine Learning*.

Xia, Y., Qin, T., Ma, W., Yu, N., and Liu, T.-Y. (2016b). Budgeted multi-armed bandits with multiple plays. In *International Joint Conference on Artificial Intelligence*.