HAL open science

Search for an immobile hider on a stochastic network

Tristan Garrec, Marco Scarsini

To cite this version:

Tristan Garrec, Marco Scarsini. Search for an immobile hider on a stochastic network. European Journal of Operational Research, 2020, 283 (2), pp.783 - 794. 10.1016/j.ejor.2019.11.040 . hal-03489798

HAL Id: hal-03489798

https://hal.science/hal-03489798v1

Submitted on 21 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0377221719309518
Manuscript_680c8580f543af436ed0240df28bcefa

Search for an Immobile Hider on a Stochastic Network

Tristan Garrec
CEREMADE, Université Paris-Dauphine
place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
& TSE-R, Toulouse School of Economics
Manufacture des Tabacs, 21 Allée de Brienne, 31015 Toulouse Cedex 6, France
tristan.garrec@ut-capitole.fr

Marco Scarsini
Dipartimento di Economia e Finanza, LUISS
Viale Romania 32, 00197 Roma, Italy
marco.scarsini@luiss.it

November 12, 2019

Abstract

Harry hides on an edge of a graph and does not move from there. Sally, starting from a known origin, tries to find him as soon as she can. Harry's goal is to be found as late as possible. At any given time, each edge of the graph is either active or inactive, independently of the other edges, with a known probability of being active. This situation can be modeled as a zero-sum two-person stochastic game. We show that the game has a value and we provide upper and lower bounds for this value. Finally, by generalizing optimal strategies of the deterministic case, we provide more refined results for trees and Eulerian graphs.

AMS Subject Classification 2010: Primary 91A24; secondary 91A05, 91A15, 91A25.

Keywords: Game theory; hide-search game; zero-sum two-person game; random graph.

© 2019 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

1 Introduction

1.1 The problem

In a typical search game a hider hides in a space and a searcher, starting from a specified point, searches for the hider, trying to find him as fast as possible. Often the space where the hider hides is assumed to be a network. In almost all existing versions of the game the network is fixed and all the edges are always available to the searcher. In real life it is often the case that some edges of the network are momentarily unavailable, for various reasons. For instance when the police are looking for a suspect in a city, it is possible that the presence of traffic, or civilians, or other unexpected obstacles, forces them to deviate from the planned path. Most often the obstacles on the network are not permanent, but vary with time. For instance, traffic may be intense in an area of the city at some time and in a different area at a different time. The vehicles involved in an accident at some point get removed from the road and traffic goes back to normal. In a more common scenario, a road may be unavailable because of a red traffic light. This scenario, although simple to describe, would require considering stochastic dependence among the availability of different edges.

Similar scenarios appear for instance when a rescue team is searching for miners in a mine. Explosions or landslides may force the rescuers to change the course of actions. Although in this case we do not have an adversarial hider, we can frame the situation as a zero-sum game, by considering the worst-case scenario, a game against Nature.

It is clear that the stochastic elements that affect the shape of the network must be taken into account by both the hider and the searcher. Consider the set of edges available to a searcher at a specific time. If the edge that she would have chosen is unavailable, she has two options: she can either wait until the edge becomes available, or she can take a different edge. Her choice clearly depends on the probability that each edge is available, on the structure of the network, and on her position in the game.

1.2 Our contribution

We study a hide-search model where a hider (Harry) hides on an edge of a graph and a searcher (Sally) travels around the graph in search of Harry. Her goal is to find him as soon as possible.

The novelty of the model is that, due to various circumstances, at any given time, some edges may be unavailable, so the graph randomly evolves over time. At each stage, each edge e of the graph is, independently of the others, active with probability p_e and inactive with probability $1 - p_e$.

At the beginning of the game, Harry hides on one edge of his choice and is immobile for the rest of the game. Starting from an initial vertex, called the root of the graph, Sally chooses at each stage a vertex among those reachable through active edges in the neighborhood of her current vertex. An equivalent approach is to let Sally choose an available incident edge, if any. The game ends when Sally traverses the edge where Harry is hidden, and his payoff is the number of stages needed for the game to end. So, Sally tries to minimize this time needed to find Harry and Harry aims to maximize this time. This can be

modeled as a zero-sum two-person game.

We first examine the deterministic version of the game when $p_e = 1$ for each edge e. This game has a value and optimal strategies. Analogously to well-known models in continuous time, we provide an upper and lower bound for this value, which correspond, for a fixed number of edges, to the value of games played on trees and on Eulerian graphs, respectively. We also characterize optimal strategies when the graph is either a tree or an Eulerian graph. We then turn to the stochastic framework and show that, even in this case, the game has a value for all positive p_e. We provide an upper and lower bound for this value and show that it converges to the value of the deterministic game when $p_e \to 1$ for each edge e. We consider some particular instances when all p_e are equal. We generalize optimal strategies of the deterministic setting to the stochastic one and obtain upper bounds on the value of the games played on binary trees and on parallel Eulerian graphs. The upper bounds are tight when Sally is restricted to some search trajectories. Finally we solve the stochastic search games played on the line and on the circle.

We are aware that the stochastic representation of reality in our model is quite simplistic, but we see this as a first step to analyze search games under uncertainty on the network structure. Moreover, even under our simplifying assumptions, the analysis is already quite complicated and general results are hard to achieve.

1.3 Related literature

Several types of hide-search games (HSGs) have been studied by various authors under different assumptions. von Neumann (1953) studied a discrete version of the model where a hider hides in a cell (i, j) of a matrix and a searcher chooses a row or column of the matrix; she finds the hider if the row or column contains the cell (i, j). The problem was framed as a two-person zero-sum game. Several variations of this discrete game were studied by various authors, among them Neuts (1963), Efron (1964), Gittins and Roberts (1979), Roberts and Gittins (1978), Sakaguchi (1973), Subelman (1981), Berry and Mensch (1986), Baston et al. (1990).

The search game with an immobile hider was introduced by Isaacs (1965). Beck and Newman (1970) considered a continuous HSG with a hider hiding on a line according to some distribution and a searcher, starting from an origin and moving at fixed speed, tries to find the hider as soon as possible. The continuous model was then generalized by Gal (1972, 1974), Gal and Chazan (1976), who, among other things extended the state space from a line to a plane.

More relevantly to our paper, some authors dealt with HSGs on a network. Among them, Bostock (1984) studied a discrete version of a continuous HSG proposed by Gal (1980). This game is played on a parallel multi-graph with three edges that join two vertices A and B and the searcher, starting from B has to find an immobile hider. The fact that the network has an odd number of parallel edges and, therefore, is not Eulerian makes the problem difficult to solve. Kikuta (1990, 1991) considered a HSG where the hider hides in one of n cells on a straight line and the searcher incurs some traveling cost. Anderson and Aramendia (1990) considered a HSG on a network and framed the problem as an infinite-dimensional linear program.

Gal (1979), Reijnierse and Potters (1993), Cao (1995), Dagan and Gal (2008), Alpern (2008) examined HSGs on trees, Eulerian networks, and some more general classes. Pavlović (1995), Gal (2000), Kikuta (2004), Alpern et al. (2008, 2009) extended the analysis to more general networks. Alpern (2011) considered a find-and-fetch game on a tree where the searcher has to find a hider on a network and can travel at speed 1 to find him, and then has to return to the origin at a different speed. Alpern and Lidbetter (2013, 2019a) replaced the usual pathwise search with what they call expanding search, where the searched area of a rooted network expands over different paths from the origin at different speeds chosen by the searcher, in such a way that the sum of the speeds is fixed. Alpern and Lidbetter (2015) dealt with a situation where the searcher can choose one of two speeds to travel and can detect the hider, when passing in front of him, only if she travels at the lower speed. Alpern (2017) considered a model where the hider can hide anywhere in a network and the searcher has to entirely traverse an edge before being able to turn around. This constraints gives the problem a more combinatorial flavor. Alpern (frth) consider a search game where the hider is constrained to hide in a fixed subset of the whole network. If this subset is the set of the midpoints of all the edges, then the model becomes similar to the one we use here, where the hider hides on edges. Related to our stochastic model, Boczkowski et al. (2018) dealt with a search model on a graph, where randomness is induced by potentially unreliable advice, that is, with some fixed probability each node is faulty and points to the wrong neighbor. von Stengel and Werchner (1997) studied the complexity of a HSG on a graph when the hider hides on one of the nodes of the graph. Jotshi and Batta (2008) proposed a heuristic algorithm to find a hider hidden uniformly at random on a network.

In the HSG studied by Alpern (2010), Alpern and Lidbetter (2014) the searcher moves on a network at a speed that depends on her location and direction. An intuitive link can be established between the speed variations considered in these two articles, and the expected time to cross some edges considered in the present article. In particular the biased depth-first strategy that we define and use in Section 6 is strictly related to the depth-first search defined in Alpern (2010).

This article should also be put into perspective with Alpern and Lidbetter (2019b) which deals with the question of knowing when depth-first search is optimal. Our article brings some elements to answer this question in the stochastic setting.

In a forthcoming paper Glazebrook et al. (2019) considered a search game where an object is hidden in one of many discrete locations and the searcher can use one of two search modes: a fast but inaccurate mode or a slow but accurate one. The reader is referred to the classical book by Alpern and Gal (2003) for an extended treatment of search games and to Hohzaki (2016) for a recent survey of the relevant literature.

To the best of our knowledge, the model where edges of a network are present only with some probability has not been studied before in the framework of search games, but is standard in other fields. For instance, it is at the foundations of the classical model of random graphs proposed by Erdős and Rényi (1959, 1960, 1961), where, given a set of vertices, a random graph is generated by creating an edge between any two pairs of vertices independently with probability p. A similar model is studied in percolation theory, where edges of a graph are independently active with probability p and one relevant problem is the num-

ber of clusters in the random graph and, as a consequence, the possibility of reaching one vertex starting from another one. The reader is referred, for instance, to Grimmett (1999), Bollobás (2001), Bollobás and Riordan (2006), van der Hofstad (2017) for a general treatment of random graphs and percolation. Bollobás et al. (2013) considered a cop and robbers games played on a random graph. Some intriguing interactions between percolation and game theory have been recently studied by Day and Falgas-Ravry (2018), Holroyd et al. (2019), who considered two-person zero-sum games on a graph with alternating moves.

1.4 Organization of the paper

The paper is organized as follows. Section 2 describes the model. Section 3 deals with the deterministic case, where all edges are active with probability 1. Section 4 shows existence of the value for the stochastic case and provides upper and lower bounds for this value. Section 5 uses dynamic programming to find best responses of the searcher against a known hiding distribution of the hider. Sections 6 and 7 are devoted to the analysis of search games on trees and Eulerian graphs, respectively. Most of the proofs can be found in Appendix A.

2 The model

2.1 Notation

Given a finite set A, we call $\operatorname{card} A$ its cardinality and $\Delta(A)$ the set of probability measures on A.

Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be a connected undirected graph, where $\mathscr{V}$ is the nonempty finite set of vertices and $\mathscr{E}$ is the nonempty finite set of edges. All edges have length 1. The degree of a vertex v is denoted $\operatorname{degr}(v)$. There exists a special vertex $O \in \mathscr{V}$, called the *root* of the graph $\mathscr{G}$. Let $\mathbb{G}$ be the set of subgraphs of $\mathscr{G}$. For all $v \in \mathscr{V}$, we call $\mathscr{N}(\mathscr{G}, v)$ the *immediate neighborhood* of v in $\mathscr{G}$:

$$\mathscr{N}(\mathscr{G}, v) = \{v\} \cup \{u \in \mathscr{V} | \{v, u\} \in \mathscr{E}\}. \tag{2.1}$$

The graph will evolve in discrete time as follows. Let $\boldsymbol{p} = (p_e)_{e \in \mathscr{E}} \in (0, 1]^{\mathscr{E}}$. At each stage $t \geq 1$, each edge $e \in \mathscr{E}$ is active with probability p_e or inactive with probability $1 - p_e$, independently of the other edges. This defines a random graph process on $\mathbb{G}$ denoted $(\mathscr{G}_t)_t = (\mathscr{V}, \mathscr{E}_t)_{t \geq 1}$, where $\mathscr{E}_t$ is the random set of active edges at time t.

2.2 The game

We consider a stochastic zero-sum game $\Gamma = \langle \mathscr{G}, O, \boldsymbol{p} \rangle$ with two players: a maximizer, called the *hider* (Harry), and a minimizer, called the *searcher* (Sally). We call this game a *stochastic search game* (SSG).

The game is played as follows. At stage 0 both players know $\mathscr{G}_0 = \mathscr{G}$ and the initial position of the searcher $v_0 = O$. The hider chooses an edge $e \in \mathscr{E}$. Then the graph $\mathscr{G}_1$ is drawn and the searcher chooses $v_1 \in \mathscr{N}(\mathscr{G}_1, v_0)$. If $\{v_0, v_1\} = e$, then the game ends and the payoff to the hider is 1, otherwise the graph $\mathscr{G}_2$

is drawn and the game continues. Inductively, at each stage $t \geq 1$, knowing $h_t = (\mathscr{G}_0, v_0, \ldots, \mathscr{G}_{t-1}, v_{t-1}, \mathscr{G}_t)$, the searcher chooses $v_t \in \mathscr{N}(\mathscr{G}_t, v_{t-1})$. If $\{v_{t-1}, v_t\} = e$, then the game ends and the payoff to the hider is t, otherwise the graph $\mathscr{G}_{t+1}$ is drawn and the game continues.

Hence in this SSG, the state space is $\mathbb{G} \times \mathscr{V}$, the action set of the hider is $\mathscr{E}$, and the action set of the searcher in state $(\mathscr{G}', v) \in \mathbb{G} \times \mathscr{V}$ is $\mathscr{N}(\mathscr{G}', v)$. We now describe the sets of strategies of the players. For $t \geq 0$, let $\mathscr{H}_t = \mathbb{G} \times (\mathbb{G} \times \mathscr{V})^t$ be the set of histories at stage t and let $\mathscr{H} = \bigcup_{t \geq 0} \mathscr{H}_t$ be the set of all histories. Call $\mathscr{S}$ the set of (behavior) strategies of the searcher, that is the strategies $\sigma \colon \mathscr{H} \to \Delta(\mathscr{V})$ such that $\sigma(h_t) \in \Delta(\mathscr{N}(\mathscr{G}_t, v_{t-1}))$.

We call pure the strategies s such that, for all $t \geq 0$ and all $h_t \in \mathscr{H}_t$,

$$s(h_t) = v_t \in \mathscr{N}(\mathscr{G}_t, v_{t-1}).$$

A behavior strategy σ naturally induces a probability measure on each $\mathscr{H}_t$, for every $t \geq 1$, which can be uniquely extended to $\mathscr{H}_\infty$ by Kolmogorov's extension theorem. This probability is denoted $\mathbb{P}_\sigma$ and the corresponding expectation is denoted $\mathbb{E}_\sigma$.

A mixed strategy of the searcher is a probability distribution over pure strategies, endowed with the product σ-algebra. By Kuhn's theorem, behavior and mixed strategies are equivalent (see, e.g., Aumann, 1964, Sorin, 2002). The sets of pure and mixed strategies of the hider are $\mathscr{E}$ and $\Delta(\mathscr{E})$, respectively. Pure strategies of the hider and the searcher will usually be denoted with the letters e and s respectively, while mixed and behavior strategies will usually be denoted with the letters ε and σ, respectively. We denote ε^{U} the uniform distribution (UD) on $\mathscr{E}$.

Finally, the payoff function of the hider is the function $g \colon \mathscr{E} \times \mathscr{S} \to \mathbb{R}_+ \cup \{+\infty\}$, defined as

$$g(e, \sigma) = \mathbb{E}_\sigma[\inf\{t \geq 1 | \{v_{t-1}, v_t\} = e\}], \tag{2.2}$$

where the infimum over the empty set is $+\infty$. The function g is linearly extended to $\Delta(\mathscr{E})$. The goal of the hider is thus to maximize the expected time by which he is found by the searcher, while the goal of the searcher is to minimize the expected time by which she finds the hider.

3 Deterministic search games

Proposition 4.1 below will show that the search game $\langle \mathscr{G}, O, \boldsymbol{p} \rangle$ has a value, which we denote $\mathrm{val}(\boldsymbol{p})$. If p_e is equal to 1 for all $e \in \mathscr{E}$, we then recover a search game with an immobile hider played on a graph. We call this game a *deterministic search game* (DSG). DSGs have a value $\mathrm{val}(\mathbf{1})$.

We recall some important definitions and results for DSGs. Versions of these results are well known when the game is played in continuous time over a continuous network (see, e.g., Alpern and Gal, 2003).

Definition 3.1. (i) A cycle in an graph is called *Eulerian* if it uses each edge exactly once. If such a cycle exists, the graph is called Eulerian.

(ii) A *Chinese postman tour* is a cycle of minimal length that visits each edge. In Eulerian graphs, the Chinese postman tours are the Eulerian cycles.

Definition 3.2. (i) The *uniform Eulerian strategy* (UES) is a mixed strategy that mixes over all Eulerian cycles with equal probability.

(ii) The *uniform Chinese postman strategy* (UCPS) is a mixed strategy that mixes over all Chinese postman tours with equal probability.

In Definition 3.2 above, note that the UCPS is not the same as a Random Chinese Postman Tour usually found in the literature. In a Random Chinese Postman Tour, the searcher follows equiprobably a Chinese postman tour or its reverse. Both strategies would be optimal in Proposition 3.4 below, however only the UCPS generalizes well to the stochastic setting.

When considering trees, we will endow them with an orientation outgoing from the root. This orientation does not affect the behavior of the searcher, who can traverse any edge in any direction, but is just needed to state and prove some of our results.

Let $\mathscr{G} = \mathscr{T}$ be a tree. If v is a vertex of $\mathscr{T}$, then $\mathscr{T}_v$ is the subtree that has v as a root and contains all edges below v in the original tree $\mathscr{T}$. Hence $\mathscr{T} = \mathscr{T}_O$.

If e is an edge of $\mathscr{G}$, then $\mathscr{T}_e := \{e\} \cup \mathscr{T}_v$ where v is the head of e, i.e., $\mathscr{T}_e$ includes e and the maximal subtree below the head of e. We denote $\mathscr{E}_v$ (resp. $\mathscr{E}_e$) the set of edges of $\mathscr{T}_v$ (resp. $\mathscr{T}_e$).

The following definition is an adaptation to our framework of what Alpern and Gal (2003, Section 3.3) have in the continuous setting.

Definition 3.3. The *equal branching distribution* (EBD) ε^* of the hider is the unique distribution on $\mathscr{E}$ that is supported on the leaf edges and, for every branching vertex v with outgoing edges $e_1, \dots, e_n$, satisfies

$$\frac{\varepsilon^*(\mathscr{E}_{e_i})}{\operatorname{card} \mathscr{E}_{e_i}} = \frac{\varepsilon^*(\mathscr{E}_{e_1})}{\operatorname{card} \mathscr{E}_{e_1}}, \quad \text{for all } i \in \{1, \dots, n\}. \tag{3.1}$$

Proposition 3.4. *Let $\Gamma = (\mathscr{V}, \mathscr{E})$. In a DSG $\Gamma = \langle \mathscr{G}, O, \mathbf{1} \rangle$ we have*

$$\operatorname{val}(\mathbf{1}) \leq \operatorname{card} \mathscr{E}. \tag{3.2}$$

Moreover, $\operatorname{val}(\mathbf{1}) = \operatorname{card} \mathscr{E}$ *if and only if $\mathscr{G}$ is a tree. In this case, the EBD and the UCPS are optimal strategies.*

We first prove the following lemma.

Lemma 3.5. *Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be a connected graph. Any Chinese postman tour has length*

(i) $2 \operatorname{card} \mathscr{E}$ *if $\mathscr{G}$ is a tree,*

(ii) *at most* $2 \operatorname{card} \mathscr{E} - 2$ *if $\mathscr{G}$ is not tree.*

Proof. If $\mathscr{G}$ is a tree the result follows by induction on $\operatorname{card}\mathscr{E}$.

Suppose now that $\mathscr{G}$ is not a tree. We again proceed by induction on $\operatorname{card}\mathscr{E}$. There exists an edge $e = \{u,v\} \in \mathscr{E}$ such that $\mathscr{G}' = (\mathscr{V}, \mathscr{E} \setminus \{e\})$ is connected.

If $\mathscr{G}'$ is a tree, we consider a Chinese postman tour $\gamma \in \mathscr{G}'$ starting at u, such that the subtree with root v is the last visited. Once the vertex v is visited for the last time on γ, we replace the end of the cycle—which has already been visited—with e, going straight from v to u. This new cycle in $\mathscr{G}$ has length at most $2(\operatorname{card}\mathscr{E} - 1) + 1 - 1 = 2\operatorname{card}\mathscr{E} - 2$, since the length of the cycle in $\mathscr{G}'$ is $2(\operatorname{card}\mathscr{E} - 1)$, the length of e is 1, and the number of the edges not visited a second time is at least 1.

If $\mathscr{G}'$ is not a tree, then it admits a Chinese postman tour γ with length at most $2(\operatorname{card}\mathscr{E} - 1) - 2$. We now consider the cycle $\gamma' \in \mathscr{G}$ which starts at u, goes back and forth on e and then follows the cycle γ on $\mathscr{G}'$. This cycle has length $2(\operatorname{card}\mathscr{E} - 1) - 2 + 2 = 2\operatorname{card}\mathscr{E} - 2$. □

The proof of Proposition 3.4 will make use of the following lemma, which refers to a model for continuous networks in continuous time. Let Q be a continuous tree network, and suppose that the edges of Q have integer length. Then Q is mapped to a tree graph $\mathscr{T}$ in the natural way. The UCPS and the EBD are defined in a similar way in $\mathscr{T}$ and in Q, and are naturally mapped from the graph setting to the continuous network setting, and vice versa.

Lemma 3.6 (Gal (1979), Alpern and Gal (2003, Theorem 3.21))**.** *Let Q be a continuous tree network with total length μ. Then*

(i) *The UCPS is an optimal search strategy.*

(ii) *The EBD is an optimal hiding strategy.*

(iii) $\operatorname{val}(\mathbf{1}) = \mu$.

If the continuous network Q with total length μ is not a tree, then $\operatorname{val}(\mathbf{1}) < \mu$.

Proof of Proposition 3.4. If $\mathscr{G}$ is a tree, the result follows from Lemma 3.6. Indeed, in the discrete setting, hiding on edges that are not leaves is strictly dominated. Similarly in the continuous setting, hiding at a point of the tree which is not terminal is strictly dominated. Hence the UCPS guarantees the value of the continuous game in the discrete one—with the natural mapping. Moreover, since the set of hiding strategies in the discrete setting is a subset of the set of hiding strategies on the continuous setting—again with the natural mapping—the EBD guarantees in the discrete game the value of the continuous one.

If $\mathscr{G}$ is not a tree, suppose that the searcher uniformly chooses between any Chinese postman tour, and let the hider choose an edge e. For any fixed Chinese postman tour of length n, e has position k in the cycle and position $n - k + 1$ in the reverse cycle. By Lemma 3.5, $n \leq 2\operatorname{card}\mathscr{E} - 2$, hence, the payoff is at most

$$\frac{k + 2\operatorname{card}\mathscr{E} - 2 - k + 1}{2} = \operatorname{card}\mathscr{E} - \frac{1}{2} < \operatorname{card}\mathscr{E}. \qquad \square$$

Proposition 3.7. *Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. In a DSG $\Gamma = \langle \mathscr{G}, O, \mathbf{1} \rangle$ we have*

$$\mathrm{val}(\mathbf{1}) \geq \frac{\mathrm{card}\,\mathscr{E} + 1}{2}. \tag{3.3}$$

Moreover, if $\mathrm{card}\,\mathscr{E} > 1$*, then*

$$\mathrm{val}(\mathbf{1}) = \frac{\mathrm{card}\,\mathscr{E} + 1}{2}. \tag{3.4}$$

if and only if $\mathscr{G}$ is Eulerian. In this case, the UD on $\mathscr{E}$ and the UES are optimal strategies.

Proof. Suppose the hider hides uniformly over $\mathscr{E}$. Now let the searcher choose any sequence of edges (without necessarily following a path in $\mathscr{G}$). Then if the searcher does not search the same edge twice during his $\mathrm{card}\,\mathscr{E}$ first picks, the payoff is $(\mathrm{card}\,\mathscr{E} + 1)/2$, hence the lower bound. Suppose $\mathrm{card}\,\mathscr{E} > 1$, it is clear that this bound is reached only in Eulerian graphs, following an Eulerian cycle, because, if the graph is not Eulerian, then an edge is visited twice. Finally, using an argument similar to the one used in Proposition 3.4, we can show that the uniform Eulerian strategy yields the payoff $(\mathrm{card}\,\mathscr{E} + 1)/2$ against any strategy of the hider. □

Together, Propositions 3.4 and 3.7 yield the next theorem, whose continuous version is a cornerstone of the search game literature. It gives bounds on the value of deterministic search games played on any graphs. Moreover, it shows that Eulerian graphs and trees are the two extreme classes of graphs in term of value of the game.

Theorem 3.8. *For any graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, the value of the DSG $\Gamma = \langle \mathscr{G}, O, \mathbf{1} \rangle$ satisfies*

$$\frac{\mathrm{card}\,\mathscr{E} + 1}{2} \leq \mathrm{val}(\mathbf{1}) \leq \mathrm{card}\,\mathscr{E}. \tag{3.5}$$

Moreover, if $\mathrm{card}\,\mathscr{E} > 1$*, the upper bound is reached if and only if $\mathscr{G}$ is a tree and the lower bound is reached if and only if $\mathscr{G}$ is an Eulerian graph.*

If $\mathscr{G}$ is an Eulerian graph, then the UD on $\mathscr{E}$ and the UES are optimal strategies.

If $\mathscr{G}$ is a tree, then the EBD and the UCPS are optimal strategies.

In Sections 6 and 7 we focus on subclasses of these two extreme classes that are Eurelian graphs and trees. Both subclasses have a recursive structure. We generalize the strategies of interest to our stochastic setting and derive bounds on the value. We also prove that these strategies are optimal in the cases of circles and lines.

4 Value of the game

Proposition 4.1. *For any $\boldsymbol{p} \in (0, 1]^{\mathscr{E}}$ the SSG $\langle \mathscr{G}, O, \boldsymbol{p} \rangle$ has a value* $\mathrm{val}(\boldsymbol{p})$*. Moreover both players have an optimal strategy.*

The proof of Proposition 4.1 is postponed to Appendix A.

Proposition 4.2. *For all $\boldsymbol{p} \in (0,1]^{\mathscr{E}}$ the value of the SSG $\langle \mathscr{G}, O, \boldsymbol{p} \rangle$ satisfies*

$$\frac{\mathrm{val}(\mathbf{1})}{1-(1-\min_{e\in\mathscr{E}} p_e)^{\delta}} \leq \mathrm{val}(\boldsymbol{p}) \leq \frac{\mathrm{val}(\mathbf{1})}{\min_{e\in\mathscr{E}} p_e}, \tag{4.1}$$

where δ is the maximum degree of $\mathscr{G}$.

As a consequence

$$\mathrm{val}(\boldsymbol{p}) \to \mathrm{val}(\mathbf{1}), \text{ as } \min_{e\in\mathscr{E}} p_e \to 1. \tag{4.2}$$

Proof. The hider guarantees the lower bound by playing as in the DSG. In expectation the searcher waits at least $(1-(1-\min_{e\in\mathscr{E}} p_e)^{\delta})^{-1}$ for a neighbor edge to be active.

We map a strategy of the searcher in the DSG to the strategy in the SSG following the same path, even if it means waiting for an edge to be active. The searcher guarantees the upper bound since it takes in expectation at most $1/\min_{e\in\mathscr{E}} p_e$ stages to cross a single edge. □

5 Dynamic programming

The next proposition is a dynamic programming formula which allows to find best responses of the searcher against a known hiding distribution of the hider. The activation parameters $\boldsymbol{p} \in (0,1]^{\mathscr{E}}$ are fixed and we omit them.

For all $\mathscr{G}_1 \in \mathbb{G}$, $v_0 \in \mathscr{V}$, $I \subset \mathscr{E}$ and $\varepsilon \in \Delta(I)$, we define

$$\mathrm{Val}(\mathscr{G}_1, v_0, I, \varepsilon) = \min_{s\in\mathscr{S}} \mathbb{E}_s \left[\sum_{e\in I} \varepsilon(e) \inf\{t \geq 1 | \{v_{t-1}, v_t\} = e\} \right]. \tag{5.1}$$

This quantity represents the value of the (one player) game in which the searcher knows the graph $\mathscr{G}_1$ and the distribution ε of the hider on $I \subset \mathscr{E}$, starts from v_0 and chooses immediately $v_1 \in \mathscr{N}(\mathscr{G}_1, v_0)$ at the first stage, before $\mathscr{G}_2$ is drawn (and then the game continues). In other words, in the true game, a graph $\mathscr{G}_1$ is drawn before Sally starts playing. Here the graph $\mathscr{G}_1$ is already fixed and Sally starts playing immediately.

Proposition 5.1. *If $I = \varnothing$, then $\mathrm{Val}(\mathscr{G}_1, v_0, I, \varepsilon) = 0$. Otherwise*

$$\mathrm{Val}(\mathscr{G}_1, v_0, I, \varepsilon) = 1 + \min_{v_1 \in \mathscr{N}(\mathscr{G}_1, v_0)} \varepsilon(I \setminus \{v_0, v_1\}) \, \mathbb{E}\left[\mathrm{Val}\left(\mathscr{G}_2, v_1, I \setminus \{v_0, v_1\}, \varepsilon^{\{v_0, v_1\}} \right) \right], \tag{5.2}$$

where $\varepsilon^{\{v_0,v_1\}}(\cdot) = \frac{1}{\varepsilon(I\setminus\{v_0,v_1\})}\varepsilon(\cdot)$, and the randomness in Eq. (5.2) is over $\mathscr{G}_2$.

Proof. If the searcher finds the hider in the first stage, which happens with probability $\varepsilon(\{v_0, v_1\})$, then the game ends and the continuation payoff is 0. On the other hand, if the searcher does not find the hider in the first stage, which happens with probability $1 - \varepsilon(\{v_0, v_1\})$, then the game continues with continuation

payoff

$$\mathbb{E}\left[\mathrm{Val}\left(\mathscr{G}_2, v_1, I \setminus \{v_0, v_1\}, \varepsilon^{\{v_0,v_1\}}\right)\right], \tag{5.3}$$

since the edge $\{v_0, v_1\}$ has been visited and the next graph $\mathscr{G}_2$ is yet to be drawn. □

6 Stochastic search games on trees

In this section and in the following one we assume

$$p_e = p \in (0, 1], \quad \text{for all } e \in \mathscr{E}. \tag{6.1}$$

Moreover in this section we assume that $\mathscr{G}$ is a tree $\mathscr{T}$ with origin O. Remark that in a tree, any strategy of the hider that consists in hiding in edges other than leaf edges is strictly dominated.

6.1 Depth-first strategies and the equal branching density

We define a particular class of strategies of the searcher in trees, called depth-first strategies. They have the property of never going backward at a vertex before having visited the whole subtree. They generalize the Chinese postman tours of the deterministic setting.

Definition 6.1. A *depth-first strategy* (DFS) on a tree is a strategy of the searcher that prescribes the following, when arriving at a vertex:

- if the set of un-searched and active outgoing edges is non-empty, take one of its edges (possibly at random);
- if all the un-searched outgoing edges are inactive, wait;
- if all outgoing edges have been searched and the backward edge is active, take it;
- if all outgoing edges have been searched and the backward edge is inactive, wait.

The *uniform depth-first strategy* (UDFS) is the DFS that, at every vertex, randomizes uniformly between all active and un-searched outgoing edges.

Definition 6.2. A DFS on $\mathscr{T}$ induces an expected time to travel from the origin O back to it, covering the entire tree. This is called the *cycle time* of $\mathscr{T}$ and is denoted $\tau(O)$. For any vertex or edge z, the cycle time of $\mathscr{T}_z$ is denoted $\tau(z)$.

Notice that $\tau(O)$ depends on $\boldsymbol{p}$, but is independent of the choice of DFS.

We now generalize Definition 3.3 to the stochastic setting, where the relevant quantity is not the number of edges of the subtrees, but rather their cycle times.

Definition 6.3. The *equal branching distribution* (EBD) ε^* of the hider is the unique distribution on the leaf edges such that, for every branching vertex v with outgoing edges $e_1, \dots, e_n$, we have

$$\frac{\varepsilon^*(\mathscr{E}_{e_i})}{\tau(e_i)} = \frac{\varepsilon^*(\mathscr{E}_{e_1})}{\tau(e_1)}, \quad \text{for all } i \in \{1, \dots, n\}. \tag{6.2}$$

Notice that Definitions 3.3 and 6.3 coincide when $p_e = 1$ for all $e \in \mathscr{E}$.

6.2 Binary trees

6.2.1 Generalities

In these sections we consider games played on binary trees, i.e., trees with at most two outgoing edges at any vertex. We call $\mathbb{T}$ the set of binary trees. DFSs allow us to obtain an upper bound for the value, when p is large enough. We also prove that this upper bound is the value of the game in which Sally is restricted to play DFSs. As a by-product we will show that, for every $p \in (0, 1]$, the UDFS and EBD are a pair of optimal strategies when the game is played on a line.

Definition 6.4. Given a tree $\mathscr{T} = (\mathscr{V}, \mathscr{E})$, we define the function $\Lambda\colon \mathbb{T} \to \mathbb{R}$ recursively as follows, where, for the sake of simplicity we use the notations $\Lambda(e) = \Lambda(\mathscr{T}_e)$ and $\Lambda(v) = \Lambda(\mathscr{T}_v)$:

If $\mathscr{T}$ has a single edge $e = (O, v)$, as in Fig. 1, then

$$\Lambda(O) = \Lambda(e) = \Lambda(v) = 0. \tag{6.3}$$

Figure 1: One edge

If $\mathrm{degr}(O) = 1$ and $e = (O, v)$, as in Fig. 2, then $\Lambda(O) = \Lambda(e) = \Lambda(v)$.

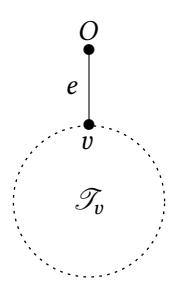

Figure 2: O has degree 1

If $\mathscr{T}$ has two edges and $\mathrm{degr}(O) = 2$, as in Fig. 3, then

$$\Lambda(O) = \frac{1}{2}\left(\frac{1}{1-(1-p)^2} - \frac{1}{p}\right). \tag{6.4}$$

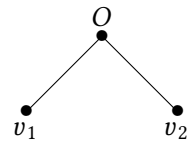

Figure 3: Two edges

If $\mathrm{degr}(O) = 2$, $e_1 = (O, v_1)$, and $e_2 = (O, v_2)$, as in Fig. 4, then

$$\Lambda(O) = \frac{\tau(v_1)}{\tau(v_1)+\tau(v_2)}\Lambda(v_1) + \frac{\tau(v_2)}{\tau(v_1)+\tau(v_2)}\Lambda(v_2) + \frac{1}{2}\left(\frac{1}{1-(1-p)^2} - \frac{1}{p}\right). \tag{6.5}$$

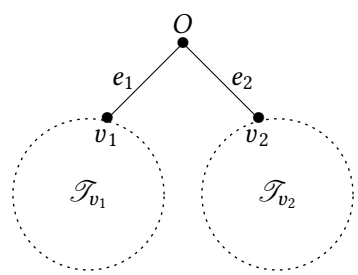

Figure 4: O has degree 2

The function Λ depends on p, but we do not make the dependence explicit.

Lemma 6.5. *Let v be a branching vertex with outgoing edges e_1 and e_2. Then for all $p \in (0, 1]$,*

$$\frac{|\Lambda(e_1)| + |\Lambda(e_2)|}{\tau(e_1)+\tau(e_2)} < \frac{1}{2}.$$

The proof of Lemma 6.5 is postponed to Appendix A. We now define the biased depth-first (behavior) strategy of the searcher.

Definition 6.6. Assume that vertex v has outgoing edges e_1 and e_2 and they are both active and unsearched. A DFS strategy σ_α is called the *biased depth-first strategy* (BDFS) if it takes e_1 with probability

$\alpha(e_1)$ and e_2 with probability $\alpha(e_2)$, where

$$\alpha(e_1) = \mathrm{proj}_{[0,1]}\left(\frac{1}{2} + \frac{\Lambda(e_1) - \Lambda(e_2)}{\tau(e_1) + \tau(e_2)} \frac{1-(1-p)^2}{p^2}\right) \tag{6.6}$$

$$\alpha(e_2) = 1 - \alpha(e_1), \tag{6.7}$$

where $\mathrm{proj}_{[0,1]}$ indicates the projection on $[0,1]$.

Theorem 6.7. *There exists $p_0 \in (0,1)$ such that for all $p \geq p_0$, the time to reach any leaf edge using the BDFS is $\frac{1}{2}\tau(O) + \Lambda(O)$. Hence for all $p \geq p_0$, we have*

$$\mathrm{val}(p) \leq \frac{1}{2}\tau(O) + \Lambda(O). \tag{6.8}$$

The proof of Theorem 6.7 is postponed to Appendix A.

Theorem 6.8. *The EBD of the hider yields the same payoff against any DFS of the searcher, and this payoff is $\frac{1}{2}\tau(O) + \Lambda(O)$.*

The proof of Theorem 6.8 is postponed to Appendix A.

Note that Definition 6.6 and Theorems 6.7 and 6.8 above have a superficial resemblance to results on the value and on biased depth-first strategies in (Alpern, 2010, Alpern and Lidbetter, 2014), where the searcher moves on a network at a speed that depends on her location and direction.

Theorems 6.7 and 6.8 imply that in a binary tree $\mathscr{G}$, if DFSs are best responses to the EBD, then there exists $p_0 \in (0,1)$ such that for all $p \geq p_0$ the value of the game is $\frac{1}{2}\tau(O) + \Lambda(O)$. Moreover the BDFS and the EBD are optimal.

Example 6.1 below is an important counterexample, as it refutes the conjecture that DFSs are best responses to the EBD.

Example 6.1. We study the game played on the tree represented in Fig. 5.

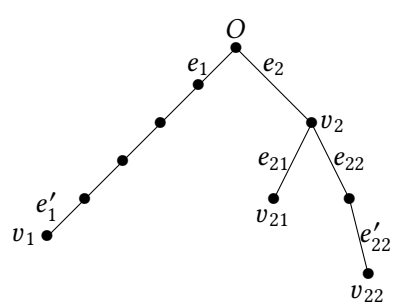

Figure 5: A counter-example

Consider the case where Sally visits v_{22} before any other leaf vertex. When she plays a DFS, this event has positive probability. Assume also that, when she has returned to v_2, after visiting v_{22}, the edge e_2 is

active but e_{21} is not. At this point she can either take edge e_2 and visit v_1 before v_{21} or wait until e_{21} becomes active and visit v_{21} before v_1. The first choice yields a lower payoff to Sally.

Indeed, visiting v_1 first yields the continuation payoff

$$g_1 = \varepsilon^*(e_1')\left(1 + \frac{5}{p}\right) + \varepsilon^*(e_{21})\left(1 + \frac{12}{p}\right),$$

whereas visiting v_{21} first yields the continuation payoff

$$g_2 = \varepsilon^*(e_{21})\left(1 + \frac{1}{p}\right) + \varepsilon^*(e_1')\left(1 + \frac{8}{p}\right).$$

The sign $g_1 - g_2$ is the same as the sign of $11\varepsilon^*(e_{21}) - 3\varepsilon^*(e_1')$, which is the same as

$$\frac{11}{3}\left(\frac{7}{p} + \frac{1}{1-(1-p)^2}\right) - \frac{30}{p},$$

which is negative for all $p \in (0, 1)$.

6.2.2 A simple binary tree

We now present a game played on a tree (Fig. 6) for which we give the value and a pair of optimal strategies for any value of $p \in (0, 1]$.

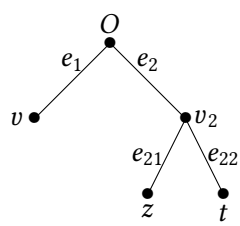

Figure 6: A simple binary tree

Let

$$p_0 = \frac{9 - \sqrt{65}}{8} \approx 0.12. \tag{6.9}$$

First case $p \geq p_0$: In this case, Sally's BDFS and Harry's EBD are a pair of optimal strategies. The value of the game is thus

$$\mathrm{val}(p) = \frac{1}{2}\tau(O) + \Lambda(O) = \frac{92 - 75p + 15p^2}{p(15 - 7p)(2 - p)}.$$

Second case $p \leq p_0$: Harry's strategy $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ is optimal. We now describe an optimal strategy of Sally.

- If no leaf edges have been visited:

- At O: if e_1 is active, take it. Otherwise, if e_2 is active but e_1 is not, take e_2.
- At v_2: take the first active edge between e_{21} and e_{22}, drawing uniformly, if they both are.

- If only e_1 has been visited, play the UDFS in the continuation game.
- If only e_{21} (resp. e_{22}) has been visited, at v_2:
  - If e_{22} (resp. e_{21}) is active, take it.
  - If e_2 is active but e_{22} (resp. e_{21}) is not, randomize, waiting at v_2 with probability $\zeta(p)$ and taking e_{22} (resp. e_{21}) with probability $1 - \zeta(p)$.
- If two leaf edges have been visited, go to the third leaf edge as quickly as possible.

The waiting probability $\zeta(p)$ is given by

$$\zeta(p) = \frac{8(2-p) - (1-p)(1+p)(2-p)}{8(2-p)(1-p) - p(1-p)^2}.$$

The value of the game is

$$\mathrm{val}(p) = \frac{1}{3}\frac{37 - 33p + 7p^2}{p(2-p)^2}.$$

6.2.3 The line

We consider a SSG played on a line. If the origin O is an extreme vertex, then the value of the game is $\mathrm{card}(\mathscr{E})/p$. We now suppose that the origin O is not an extreme vertex, and that the line has $L = \lambda_1 + \lambda_2$ edges (λ_1 on the left side of O and λ_2 on the right side) as shown in Fig. 7.

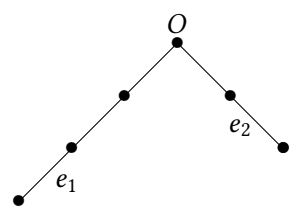

Figure 7: The line with $\lambda_1 = 3$ and $\lambda_2 = 2$

In this case, for all $p \in (0, 1]$ the BDFS is the UDFS σ^*, and the EBD of the hider is

$$\varepsilon^* = \left(\frac{\lambda_1}{L}, \frac{\lambda_2}{L}\right).$$

Proposition 6.9. *If the graph $\mathscr{G}$ is a line, then DFS are best responses to the EBD. Hence, $(\varepsilon^*, \sigma^*)$ is a pair of optimal strategies.*

Proof. Harry plays ε^*. At O, whatever active edge Sally takes, the continuation payoff is $(\lambda_1 + \lambda_2 - 1)/p$. Hence she does not profit from waiting for one specific edge to be active. □

Together with Theorems 6.7 and 6.8, Proposition 6.9 yields the following corollary.

Corollary 6.10. *The value of the game played on the line with L edges is*

$$\mathrm{val}(p) = \frac{1}{2}\tau(O) + \Lambda(O) = \frac{L}{p} + \frac{1}{1-(1-p)^2} - \frac{1}{p},$$

for all $p \in (0, 1]$, if the root is not an extreme vertex. Moreover the EBD and the UDFS are optimal strategies.

7 Stochastic search games on Eulerian graphs

7.1 Eulerian strategies and the uniform density

For Eulerian graphs we define a strategy of the searcher, called *Eulerian strategy* (ES), which generalizes an Eulerian cycle of the deterministic setting. At any vertex an ES chooses an active outgoing edge that had not previously been visited in such a way that the induced path is an Eulerian cycle. The ES that at any vertex randomizes uniformly over the outgoing edges is called the *uniform Eulerian strategy* (UES) and is denoted σ^*.

Definition 7.1. The UES on a Eulerian graph $\mathscr{G}$ induces an expected time to travel from the origin O covering the entire Eulerian graph. This is called the *cycle time* of $\mathscr{G}$ and is denoted $\theta(\mathscr{G})$.

7.2 Parallel Eulerian graphs

7.2.1 Generalities

We call *parallel graph* a graph where parallel paths link two vertices, one of these two vertices being the root O, as in Fig. 8. Such a graph is denoted $\mathscr{P}_n(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ is the vector of the lengths of the parallel paths. The parallel uniform strategy of Sally consists in choosing at O uniformly between active and unsearched edges and then going straight to D on the current parallel path (and similarly at D).

Remark that if the number of parallel paths $n = 2m$ is even, then the parallel graph is Eulerian and we call it a *parallel Eulerian graph*. In this case, the parallel uniform strategy is the UES. For a parallel Eulerian graph $\mathscr{P}_{2m}(\boldsymbol{\lambda})$ with $2m$ parallel lines, the cycle time of $\mathscr{P}_{2m}(\boldsymbol{\lambda})$ is

$$\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) = \sum_{k=1}^{2m} \left(\frac{1}{1-(1-p)^k} + \frac{\lambda_k - 1}{p} \right).$$

The UES allows us to obtain an upper bound for the value. We also prove that this upper bound is the value of the game in which Sally is restricted to play ESs. As a by-product we will show that, for every $p \in (0, 1]$, the UES and UD are a pair of optimal strategies when the game is played on a circle.

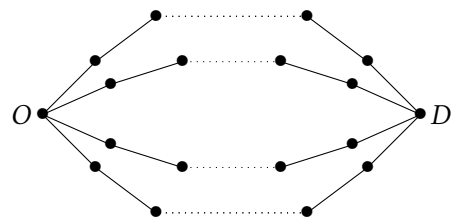

Figure 8: A parallel Eulerian graph

Definition 7.2. Given a parallel Eulerian graph $\mathscr{P}_{2m}(\boldsymbol{\lambda})$ with $2m$ parallel lines, let Φ_m be the following quantity defined recursively:

$$\Phi_1 = \frac{1}{2}\left(\frac{1}{1-(1-p)^2} - \frac{1}{p}\right), \tag{7.1}$$

and for eack $m > 1$,

$$\begin{aligned}\Phi_m = \frac{1}{2}\frac{1}{1-(1-p)^{2m}} + \left(\frac{1}{2} - \frac{1}{2m}\right)\frac{1}{1-(1-p)^{2m-1}} \\ - \frac{1}{2m}\left(\sum_{k=1}^{2(m-1)} \frac{1}{1-(1-p)^k} + \frac{1}{p}\right) + \frac{m-1}{m}\Phi_{m-1}.\end{aligned}$$

Remark that Φ_m only depends on the number of parallel paths and not on their length.

Theorem 7.3. *On a parallel Eulerian graph $\mathscr{P}_{2m}(\boldsymbol{\lambda})$, the expected time to reach any edge using the UES is*

$$\frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m. \tag{7.2}$$

Hence, for all $p \in (0,1]$, we have

$$\mathrm{val}(p) \leq \frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m. \tag{7.3}$$

The proof of Theorem 7.3 is postponed to Appendix A.

Theorem 7.4. *On a parallel Eulerian graph $\mathscr{P}_{2m}(\boldsymbol{\lambda})$, the uniform density of the hider yields the same payoff*

$$\frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m$$

against any Eulerian strategy of the searcher.

The proof of Theorem 7.4 is postponed to Appendix A. Theorems 7.3 and 7.4 imply that in a parallel Eulerian graph $\mathscr{P}_{2m}(\boldsymbol{\lambda})$, if Eulerian strategies are best responses to the uniform density, for all $p \in (0,1]$

the value of the game is

$$\frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m.$$

Moreover the UES and the UD are optimal.

However, Eulerian strategies are not always best responses to the UD, as we now argue.

Example 7.1. We study the game played on a parallel Eulerian graph with four parallel paths. Each path i has two edges $e_{i1} = \{O, v_i\}$ and $e_{i2} = \{v_i, D\}$, where v_i is the middle vertex of the i-th path.

Consider the case where Sally visits e_{41}, e_{42} and e_{12} before any other edge. When she plays an ES, this event has positive probability. Assume also that, when at v_1, the edge e_{12} is active but e_{11} is not. At this point she can either wait at v_2 until e_{11} becomes active in order to follow an ES, or she can take e_{12}, then the first active edge between e_{22} and e_{32} and continue with e_{21} or e_{31} respectively. Finally, she takes the first active edge between e_{11} and the other edge at O that has not been visited yet, and then visits the two remaining edges as quickly as possible.

Following an ES yields the continuation payoff

$$g_1 = \frac{1}{5}\left(5 + \frac{11}{p} + \frac{4}{1-(1-p)^2}\right).$$

Following the second strategy yields the continuation payoff

$$g_2 = \frac{1}{5}\left(5 + \frac{17}{2p} + \frac{8}{1-(1-p)^2}\right).$$

Hence if $p < 2/5$, the second strategy yields a lower payoff to Sally than an ES.

7.2.2 The circle

We now examine the game played on a circle.

Lemma 7.5. *If the graph $\mathscr{G}$ is a circle, then Eulerian strategies are best responses to the uniform density.*

The proof of Lemma 7.5 is rather straightforward and we omit it. Together with Theorems 7.3 and 7.4, Lemma 7.5 yields the following corollary.

Corollary 7.6. *The value of the game played on the circle with L edges is*

$$\mathrm{val}(p) = \frac{\theta(\mathscr{G}) + p^{-1}}{2} + \Phi_2 = \frac{1}{1-(1-p)^2} + \frac{L-1}{2p},$$

for all $p \in (0, 1]$. Moreover the uniform density and the uniform Eulerian strategy are optimal strategies.

Acknowledgments

This work has been partly supported by COST Action CA16228 European Network for Game Theory. Marco Scarsini is a member of GNAMPA-INdAM. Tristan Garrec gratefully acknowledges the hospitality of the Department of Economics and Finance at LUISS, where part of this research was carried out.

A Omitted proofs

A.1 Omitted proofs of Section 4

The following lemma is a corollary of Flesch et al. (2019, Theorem 12).

Lemma A.1. *Positive zero-sum stochastic games with finite state space and action spaces have a value. Moreover the minimizer has an optimal (stationary) strategy.*

Proof of Proposition 4.1. We restate the stochastic search game as a positive zero-sum stochastic game with finite state and action spaces and apply Lemma A.1. The idea is that Sally's stage payoff is 1 at each stage until she finds Harry, transitioning then to an absorbing state in which the payoff is 0 forever. The total payoff is then the sum of the stage payoffs.

Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be the underlying graph. In order to cast our problem in the framework of Flesch et al. (2019), we will use a finite state space $(\mathscr{V} \times \mathbb{G} \times (\mathscr{E} \cup \{\dagger\})) \cup \{*\}$, which is larger than the state space $\mathbb{G} \times \mathscr{V}$, used in Section 2. The state $(v_0, (\mathscr{V}, \varnothing), \dagger)$ is the initial state at stage 0, where $\dagger$ indicated that the hider has not chosen an edge where to hide. In this state, the finite action space of the searcher is $\mathscr{N}((\mathscr{V}, \varnothing), v_0) = \{v_0\}$, the finite action space of the hider is $\mathscr{E}$ and the payoff is 0. The state $*$ is an absorbing state in which the payoff is 0 forever. In any other state the payoff is 1.

The state moves from the initial state to $(v_0, \mathscr{G}_1, e)$ where $\mathscr{G}_1$ is the graph drawn at stage 1 and e is the edge chosen by the hider (which is fixed for the rest of the game). In any state $(v, \mathscr{G}', e) \in \mathscr{V} \times \mathbb{G} \times \mathscr{E}$ the searcher selects $v' \in \mathscr{N}(\mathscr{G}', v)$ and the hider selects $e \in \{e\}$. If $\{v, v'\} = e$ then the state next moves to the absorbing state $*$. If $\{v, v'\} \neq e$, the state moves to $(v', \mathscr{G}'', e)$ where $\mathscr{G}''$ is drawn according to the activation parameter.

Finally, since $\mathscr{E}$ is finite, the hider has an optimal strategy. □

A.2 Omitted proofs of Section 6

Proof of Lemma 6.5. We proceed by induction on the number of edges. The base case is immediate since $\Lambda(e_1) = \Lambda(e_2) = 0$. For the induction step the situation is represented in Fig. 9. The vertex v_1 is the first vertex encountered in $\mathscr{T}_{e_1}$ with two outgoing edges, and similarly for v_2 and $\mathscr{T}_{e_2}$.

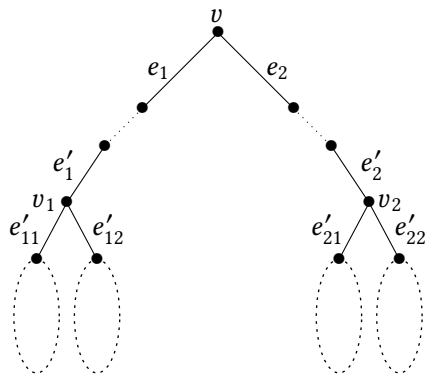

Figure 9: The induction step

We have

$$\begin{aligned}|\Lambda(e_1)| = |\Lambda(e'_1)| &= \left|\frac{\tau(e'_{11})}{\tau(e'_{11})+\tau(e'_{12})}\Lambda(e'_{11}) + \frac{\tau(e'_{12})}{\tau(e'_{11})+\tau(e'_{12})}\Lambda(e'_{12}) + \frac{1}{2}\left(\frac{1}{1-(1-p)^2} - \frac{1}{p}\right)\right| \\ &< \frac{1}{2}\max(\tau(e'_{11}), \tau(e'_{12})) + \frac{1}{2}\left|\frac{1}{1-(1-p)^2} - \frac{1}{p}\right|\end{aligned}$$

by induction, and similarly for $\Lambda(e_2)$. Moreover we have

$$\tau(e_1) > \tau(e'_1) = \tau(e'_{11}) + \tau(e'_{12}) + \frac{1}{p} + \frac{1}{1-(1-p)^2},$$

and similarly for $\tau(e_2)$. Finally,

$$\begin{aligned}\frac{|\Lambda(e_1)| + |\Lambda(e_2)|}{\tau(e_1)+\tau(e_2)} &< \frac{\frac{1}{2}\left(\max(\tau(e'_{11}), \tau(e'_{12})) + \max(\tau(e'_{21}), \tau(e'_{22})) + \frac{2}{p} - \frac{2}{1-(1-p^2)}\right)}{\tau(e'_{11}) + \tau(e'_{12}) + \tau(e'_{21}) + \tau(e'_{22}) + \frac{2}{p} + \frac{2}{1-(1-p^2)}} \\ &< \frac{1}{2}.\end{aligned}$$

□

Proof of Theorem 6.7. We proceed by induction on the number of edges in the tree $\mathscr{T}$. If $\mathscr{T}$ has only one edge e, then

$$g(e, \sigma_\alpha) = \frac{1}{p} = \frac{1}{2}\left(\frac{2}{p} + 0\right). \tag{A.1}$$

Suppose that for any tree that has less edges than $\mathscr{T}$, the time to reach any leaf edge using the BDFS is $\frac{1}{2}\tau(O) + \Lambda(O)$.

If the origin O has degree 1 (as in Fig. 2), then, for any leaf edge e, we have

$$g(e, \sigma_\alpha) = \frac{1}{p} + \frac{1}{2}(\tau(v)) + \Lambda(v) = \frac{1}{2}\left(\tau(v) + \frac{2}{p}\right) + \Lambda(v) = \frac{1}{2}\tau(O) + \Lambda(O). \tag{A.2}$$

Consider the case where O has degree 2 (as in Fig. 4) and let e_1 be a leaf edge in $\mathscr{T}_{v_1}$. Then

$$\begin{aligned} g(e_1,\sigma_\alpha) &= (1-p)^2(1+g(e_1,\sigma_\alpha)) \\ &\quad + p(1-p)\left(1+\frac{1}{2}\tau(v_1)+\Lambda(v_1)+1+\tau(v_2)+\frac{2}{p}+\frac{1}{2}\tau(v_1)+\Lambda(v_1)\right) \\ &\quad + p^2\left(\alpha(e_1)\left(1+\frac{1}{2}\tau(v_1)+\Lambda(v_1)\right)+\alpha(e_2)\left(1+\tau(v_2)+\frac{2}{p}+\frac{1}{2}\tau(v_1)+\Lambda(v_1)\right)\right). \end{aligned}$$

and

$$\begin{aligned} g(e_1,\sigma_\alpha)(1-(1-p)^2) &= 1+p(1-p)\left(\tau(v_1)+\tau(v_2)+\frac{2}{p}+2\Lambda(v_1)\right) \\ &\quad + p^2\left(\frac{1}{2}\tau(v_1)+\Lambda(v_1)+\alpha(e_2)\left(\tau(v_2)+\frac{2}{p}\right)\right). \end{aligned}$$

Furthermore,

$$\tau(O) = \tau(v_1)+\tau(v_2)+\frac{3}{p}+\frac{1}{1-(1-p)^2}$$

and

$$\tau(e_1)+\tau(e_2) = \tau(v_1)+\tau(v_2)+\frac{4}{p} = \tau(O)+\frac{1}{p}-\frac{1}{1-(1-p)^2}.$$

Hence, by Lemma 6.5, for p large enough we do not need the projection in Eq. (6.6), so we have

$$\begin{aligned} g(e_1,\sigma_\alpha)(1-(1-p)^2) &= 1+p(1-p)\left(\tau(O)-\frac{1}{p}-\frac{1}{1-(1-p)^2}+2\Lambda(e_1)\right) \\ &\quad + p^2\left(\frac{1}{2}\left(\tau(e_1)-\frac{2}{p}\right)+\Lambda(e_1)+\left(\frac{1}{2}+\frac{\Lambda(e_2)-\Lambda(e_1)}{\tau(e_1)+\tau(e_2)}\frac{1-(1-p)^2}{p^2}\right)\tau(e_2)\right). \end{aligned}$$

Thus,

$$\begin{aligned} g(e_1,\sigma_\alpha) &= \frac{1}{1-(1-p)^2}+\frac{1}{2}\left(\tau(O)-\frac{1}{p}-\frac{1}{1-(1-p)^2}+2\Lambda(e_1)\right)+\frac{\Lambda(e_2)-\Lambda(e_1)}{\tau(e_1)+\tau(e_2)}\tau(e_2) \\ &= \frac{1}{2}\tau(O)+\frac{1}{2}\left(\frac{1}{1-(1-p)^2}-\frac{1}{p}\right)+\Lambda(e_1)\frac{\tau(e_1)}{\tau(e_1)+\tau(e_2)}+\Lambda(e_2)\frac{\tau(e_2)}{\tau(e_1)+\tau(e_2)} \\ &= \frac{1}{2}\tau(O)+\Lambda(O). \end{aligned}$$

□

Proof of Theorem 6.8. The proof is by induction on the number of edges of the tree $\mathscr{T}$. If $\mathscr{T}$ has only one edge, the result is immediate. Suppose now that the results holds for any tree with fewer edges than $\mathscr{T}$.

If the degree of the origin O is 1, the result follows immediately from the induction hypothesis. Assume now that the degree of O is 2 (as in Fig. 4). Let $s(v_1)$ and $s(v_2)$ be two DFSs on $\mathscr{T}_{v_1}$ and $\mathscr{T}_{v_2}$, respectively.

Let $s(e_1)$ be the pure DFS on $\mathscr{T}$ that, when both e_1 and e_2 are active, takes edge e_1 concatenated with $s(v_1)$ and then $s(v_2)$, in case Harry is not found in $\mathscr{T}_{v_1}$. The pure strategy $s(e_2)$ is defined analogously. Given a vertex v, call ε_v^* the conditional probability measure on $\mathscr{E}_v$ induced by ε^*. Then

$$\begin{aligned} g(\varepsilon^*, s(e_1)) = {} & (1-p)^2(1+g(\varepsilon^*, s(e_1))) \\ & + p\left(\varepsilon^*(\mathscr{E}_{e_1})(1+g(\varepsilon_{v_1}^*, s(v_1)) + \varepsilon^*(\mathscr{E}_{e_2})\left(1+\frac{2}{p}+\tau(v_1)+g(\varepsilon_{v_2}^*, s(v_2))\right)\right) \\ & + p(1-p)\left(\varepsilon^*(\mathscr{E}_{e_2})(1+g(\varepsilon_{v_2}^*, s(v_2)) + \varepsilon^*(\mathscr{E}_{e_1})\left(1+\frac{2}{p}+\tau(v_2)+g(\varepsilon_{v_1}^*, s(v_1))\right)\right). \end{aligned}$$

Hence,

$$\begin{aligned} g(\varepsilon^*, s(e_1)) = g(\varepsilon^*, s(e_2)) &\iff \varepsilon^*(\mathscr{E}_{e_1})\left(\frac{2}{p}+\tau(v_2)\right) = \varepsilon^*(\mathscr{E}_{e_2})\left(\frac{2}{p}+\tau(v_1)\right) \\ &\iff \varepsilon^*(\mathscr{E}_{e_1}) = \frac{\tau(e_1)}{\tau(e_1)+\tau(e_2)}. \end{aligned}$$

□

A.3 Omitted proofs of Section 7

Proof of Theorem 7.3. We denote $e(i,j)$ the j-th edge of path i, starting from the root O. We proceed by induction on m.

Consider that, with probability $(1-p)^{2m}$ all edges starting from O are inactive; if this happens, Sally has to wait one turn and her payoff is $(1+g(e(i,j),\sigma^*))$. With probability $1-(1-p)^{2m}$ at least one edge is active and each of the available edges is chosen with equal probability. Given that Harry hides in $e(i,j)$, if the chosen path is i, then the game ends in $(j-1)/p$ units of time. If the chosen path is $k \neq i$, then Sally goes to D and the continuation payoff is $g_k(e(i,\lambda_i - j+1),\sigma^*)$, where g_k is the payoff of the game played on $\mathscr{P}_{2m-1}(\boldsymbol{\lambda} \setminus \lambda_k)$, in which path k has been visited, $\boldsymbol{\lambda} \setminus \lambda_k$ is the vector $(\lambda_1, \ldots, \lambda_{k-1}, \lambda_{k+1}, \ldots, \lambda_{2m})$ of size $2m-1$, and the game starts in D.

In formula:

$$\begin{aligned} g(e(i,j),\sigma^*) = {} & (1-p)^{2m}(1+g(e(i,j),\sigma^*)) \\ & + \frac{1-(1-p)^{2m}}{2m}\left(1+\frac{j-1}{p}+\sum_{k\neq i}\left(1+\frac{\lambda_k-1}{p}+g_k(e(i,\lambda_i-j+1),\sigma^*)\right)\right). \end{aligned}$$

The above expression yields

$$g(e(i,j),\sigma^*) = \frac{1}{1-(1-p)^{2m}} + \frac{1}{2m}\left(\frac{j-1}{p}+\sum_{k\neq i}\left(\frac{\lambda_k-1}{p}+g_k(e(i,\lambda_i-j+1),\sigma^*)\right)\right) \tag{A.3}$$

A similar expression holds for $g_k(e(i, \lambda_i - j + 1), \sigma^*)$. Plugging it in Eq. (A.3), we obtain

$$\begin{aligned} g(e(i,j),\sigma^*) =& \frac{1}{1-(1-p)^{2m}} + \frac{1}{2m}\left(\frac{j-1}{p} + \sum_{k\neq i}\left(\frac{\lambda_k - 1}{p} + \frac{1}{1-(1-p)^{2m-1}}\right.\right. \\ &\left.\left. + \frac{1}{2m-1}\left(\frac{\lambda_i - j}{p} + \sum_{k'\neq k,i}\left(\frac{\lambda_{k'}-1}{p} + g_{k,k'}(e(i,j),\sigma^*)\right)\right)\right)\right), \end{aligned} \tag{A.4}$$

where $g_{k,k'}$ is the payoff of the game played on $\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda} \setminus \lambda_k, \lambda_{k'})$, in which both path k and path k' have been visited. The induction hypothesis is

$$g_{k,k'}(e(i,j),\sigma^*) = \frac{\theta(\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda} \setminus \lambda_k, \lambda_{k'})) + p^{-1}}{2} + \Phi_{m-1}. \tag{A.5}$$

Therefore, plugging Eq. (A.5) into Eq. (A.4), we get

$$\begin{aligned} g(e(i,j),\sigma^*) =& \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} \\ &+ \frac{1}{2m}\left(1 + \frac{2m-2}{2m-1}\right)\sum_{k\neq i}\frac{\lambda_k - 1}{p} + \frac{1}{2m}\frac{\lambda_i - 1}{p} \\ &+ \frac{1}{2m(2m-1)}\sum_{k\neq i}\sum_{k'\neq i,k}\left(\frac{\theta(\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda} \setminus \lambda_k, \lambda_{k'})) + p^{-1}}{2} + \Phi_{m-1}\right). \end{aligned}$$

Furthermore,

$$\begin{aligned} \sum_{k\neq i}\sum_{k'\neq i,k}\left(\theta(\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda} \setminus \lambda_k, \lambda_{k'})) + p^{-1}\right) =& \frac{(2m-1)(2m-2)}{p} \\ &+ (2m-1)(2m-2)\sum_{k=1}^{2(m-1)}\frac{1}{1-(1-p)^k} \\ &+ (2m-1)(2m-2)\frac{\lambda_i - 1}{p} \\ &+ (2m-2)(2m-3)\sum_{k\neq i}\frac{\lambda_k - 1}{p}. \end{aligned}$$

And finally, one obtains the following simplifications

$$
\begin{aligned}
g(e(i,j),\sigma^*) &= \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} + \frac{2m-2}{4m}\left(\frac{1}{p} + \sum_{k=1}^{2(m-1)} \frac{1}{1-(1-p)^k}\right) \\
&\quad + \frac{1}{2m}\left(1+\frac{2m-2}{2m-1}\right)\sum_{k\neq i}\frac{\lambda_k-1}{p} + \frac{1}{2m}\frac{\lambda_i-1}{p} + \frac{2m-2}{4m}\frac{\lambda_i-1}{p} \\
&\quad + \frac{(2m-2)(2m-3)}{4m(2m-1)}\sum_{k\neq i}\frac{\lambda_k-1}{p} + \frac{2m-2}{2m}\Phi_{m-1} \\
&= \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} \\
&\quad + \frac{m-1}{2m}\left(\frac{1}{p} + \sum_{k=1}^{2(m-1)} \frac{1}{1-(1-p)^k}\right) + \frac{1}{2}\sum_{k=1}^{2m}\frac{\lambda_k-1}{p} + \frac{m-1}{m}\Phi_{m-1} \\
&= \frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m.
\end{aligned}
$$

□

Proof of Theorem 7.4. The proof is by induction on the number of parallel paths. Let s be a ES of Sally, and denote

$$L = \sum_{k=1}^{2m} \lambda_k$$

the number of edges of $\mathscr{P}_{2m}(\boldsymbol{\lambda})$. First,

$$g(\varepsilon^{\mathrm{U}}, s) = \frac{1}{1-(1-p)^{2m}} + \frac{1}{2m}\left(1-\frac{1}{L}\right)\sum_{k=1}^{2m} g^{\lambda_k-1}(\varepsilon^{\mathrm{U}}, s),$$

where $g^{\lambda_k-1}(\varepsilon^{\mathrm{U}}, s)$ is the payoff of the continuation game after one edge of path k has been visited. It is not difficult to prove that

$$(L-1)g^{\lambda_k-1}(\varepsilon^{\mathrm{U}}, s) = (L-\lambda_i)g_k(\varepsilon^{\mathrm{U}}, s) + \frac{L(\lambda_k-1)}{p} - \frac{\lambda_k(\lambda_k-1)}{2p},$$

where g_k is the payoff of the game played on $\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda}\setminus\lambda_k)$, in which path k has been visited. Therefore

$$g(\varepsilon^{\mathrm{U}}, s) = \frac{1}{1-(1-p)^{2m}} + \frac{L}{2mp} - \frac{1}{p} + \frac{1}{2mL}\sum_{k=1}^{2m}\left(\frac{-\lambda_k(\lambda_k-1)}{2p} + (L-\lambda_k)g_k(\varepsilon^{\mathrm{U}}, s)\right).$$

Computing a similar expression for $g_k(\varepsilon^{\mathrm{U}}, s)$ and plugging it in the above equation one has

$$\begin{aligned} g(\varepsilon^{\mathrm{U}}, s) = {} & \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} + \frac{L}{2mp} - \frac{1}{p} - \frac{2m-1}{2mp} \\ & + \frac{1}{2mL}\sum_{k=1}^{2m}\left(\frac{-\lambda_k(\lambda_k-1)}{2p} + \frac{(L-\lambda_k)^2}{(2m-1)p}\right. \\ & \left. + \frac{1}{(2m-1)}\sum_{k'\neq k}\left(\frac{-\lambda_{k'}(\lambda_{k'}-1)}{2p} + (L-\lambda_k-\lambda_{k'})g_{k,k'}(\varepsilon^{\mathrm{U}}, s)\right)\right), \end{aligned}$$

where $g_{k,k'}$ is the payoff of the game played on $\mathscr{P}_{2(m-1)}(\boldsymbol{\lambda} \setminus \lambda_k, \lambda_{k'})$, in which both path k and path k' have been visited. From the induction hypothesis, one has

$$g_{k,k'}(\varepsilon^{\mathrm{U}}, s) = \frac{1}{2}\left(\frac{1}{p} + \frac{L-\lambda_k-\lambda_{k'}-2(m-1)}{p} + \sum_{l=1}^{2(m-1)}\frac{1}{1-(1-p)^l}\right) + \Phi_{m-1}.$$

Plugging this expression in the previous equation, one has

$$\begin{aligned} g(\varepsilon^{\mathrm{U}}, s) = {} & \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} + \frac{L}{2mp} - \frac{1}{p} - \frac{2m-1}{2mp} + \frac{m-1}{m}\Phi_{m-1} \\ & + \frac{m-1}{2m}\left(\frac{1}{p} - \frac{2(m-1)}{p} + \sum_{l=1}^{2(m-1)}\frac{1}{1-(1-p)^l}\right) \\ & + \frac{1}{2mL}\sum_{k=1}^{2m}\left(\frac{-\lambda_k(\lambda_k-1)}{2p} + \frac{(L-\lambda_k)^2}{(2m-1)p} + \frac{1}{2m-1}\sum_{k'\neq k}\left(\frac{-\lambda_{k'}(\lambda_{k'}-1)}{2p} + \frac{(L-\lambda_k-\lambda_{k'})^2}{2p}\right)\right). \end{aligned}$$

Furthermore

$$\begin{aligned} & \frac{1}{2mL}\sum_{k=1}^{2m}\left(\frac{-\lambda_k(\lambda_k-1)}{2p} + \frac{(L-\lambda_k)^2}{(2m-1)p} + \frac{1}{2m-1}\sum_{k'\neq k}\left(\frac{-\lambda_{k'}(\lambda_{k'}-1)}{2p} + \frac{(L-\lambda_k-\lambda_{k'})^2}{2p}\right)\right) \\ & \qquad = \frac{1}{2mp} + \frac{(m-1)L}{2mp}. \end{aligned}$$

Finally, we have

$$\begin{aligned} g(\varepsilon^{\mathrm{U}}, s) &= \frac{1}{1-(1-p)^{2m}} + \frac{2m-1}{2m}\frac{1}{1-(1-p)^{2m-1}} + \frac{m-1}{2m}\left(\frac{1}{p} + \sum_{k=1}^{2(m-1)}\frac{1}{1-(1-p)^k}\right) \\ &\quad + \frac{1}{2}\sum_{k=1}^{2m}\frac{\lambda_k-1}{p} + \frac{m-1}{m}\Phi_{m-1} \\ &= \frac{\theta(\mathscr{P}_{2m}(\boldsymbol{\lambda})) + p^{-1}}{2} + \Phi_m. \end{aligned}$$

□

References

Alpern, S. (2008). Hide-and-seek games on a tree to which Eulerian networks are attached. *Networks*, 52(3):162–166.

Alpern, S. (2010). Search games on trees with asymmetric travel times. *SIAM J. Control Optim.*, 48(8):5547–5563.

Alpern, S. (2011). Find-and-fetch search on a tree. *Oper. Res.*, 59(5):1258–1268.

Alpern, S. (2017). Hide-and-seek games on a network, using combinatorial search paths. *Oper. Res.*, 65(5):1207–1214.

Alpern, S. (frth). Search for an immobile hider in a known subset of a network. *Theoret. Comput. Sci.* forthcoming.

Alpern, S., Baston, V., and Gal, S. (2008). Network search games with immobile hider, without a designated searcher starting point. *Internat. J. Game Theory*, 37(2):281–302.

Alpern, S., Baston, V., and Gal, S. (2009). Searching symmetric networks with utilitarian-postman paths. *Networks*, 53(4):392–402.

Alpern, S. and Gal, S. (2003). *The Theory of Search Games and Rendezvous*. Kluwer Academic Publishers, Boston, MA.

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

Alpern, S. and Lidbetter, T. (2014). Searching a variable speed network. *Math. Oper. Res.*, 39(3):697–711.

Alpern, S. and Lidbetter, T. (2015). Optimal trade-off between speed and acuity when searching for a small object. *Oper. Res.*, 63(1):122–133.

Alpern, S. and Lidbetter, T. (2019a). Approximate solutions for expanding search games on general networks. *Ann. Oper. Res.*, 275(2):259–279.

Alpern, S. and Lidbetter, T. (2019b). Search and delivery man problems: When are depth-first paths optimal? *arXiv:1910.13178*.

Anderson, E. J. and Aramendia, M. A. (1990). The search game on a network with immobile hider. *Networks*, 20(7):817–844.

Aumann, R. J. (1964). Mixed and behavior strategies in infinite extensive games. In *Advances in Game Theory*, pages 627–650. Princeton Univ. Press, Princeton, N.J.

Baston, V. J., Bostock, F. A., and Ruckle, W. H. (1990). The gold-mine game. *J. Optim. Theory Appl.*, 64(3):641–650.

Beck, A. and Newman, D. J. (1970). Yet more on the linear search problem. *Israel J. Math.*, 8:419–429.

Berry, D. A. and Mensch, R. F. (1986). Discrete search with directional information. *Oper. Res.*, 34(3):470–477.

Boczkowski, L., Korman, A., and Rodeh, Y. (2018). Searching a tree with permanently noisy advice. In *26th European Symposium on Algorithms*, volume 112 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 54, 13. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern.

Bollobás, B. (2001). *Random Graphs*. Cambridge University Press, Cambridge, second edition.

Bollobás, B., Kun, G., and Leader, I. (2013). Cops and robbers in a random graph. *Journal of Combinatorial Theory, Series B*, 103:226–236.

Bollobás, B. and Riordan, O. (2006). *Percolation*. Cambridge University Press, New York.

Bostock, F. A. (1984). On a discrete search problem on three arcs. *SIAM J. Algebraic Discrete Methods*, 5(1):94–100.

Cao, B. (1995). Search-hide games on trees. *European J. Oper. Res.*, 80(1):175–183.

Dagan, A. and Gal, S. (2008). Network search games, with arbitrary searcher starting point. *Networks*, 52(3):156–161.

Day, A. N. and Falgas-Ravry, V. (2018). Maker-breaker percolation games I: crossing grids. arXiv:1810.05190.

Efron, B. (1964). Optimum evasion versus systematic search. *J. Soc. Indust. Appl. Math.*, 12:450–457.

Erdős, P. and Rényi, A. (1959). On random graphs. I. *Publ. Math. Debrecen*, 6:290–297.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61.

Erdős, P. and Rényi, A. (1961). On the evolution of random graphs. *Bull. Inst. Internat. Statist.*, 38:343–347.

Flesch, J., Predtetchinski, A., and Sudderth, W. (2019). Positive zero-sum stochastic games with countable state and action spaces. *Appl. Math. Optim.*, forthcoming.

Gal, S. (1972). A general search game. *Israel J. Math.*, 12:32–45.

Gal, S. (1974). A discrete search game. *SIAM J. Appl. Math.*, 27:641–648.

Gal, S. (1979). Search games with mobile and immobile hider. *SIAM J. Control Optim.*, 17(1):99–122.

Gal, S. (1980). *Search games.* Academic Press, Inc., New York-London.

Gal, S. (2000). On the optimality of a simple strategy for searching graphs. *Internat. J. Game Theory*, 29(4):533–542 (2001).

Gal, S. and Chazan, D. (1976). On the optimality of the exponential functions for some minimax problems. *SIAM J. Appl. Math.*, 30(2):324–348.

Gittins, J. C. and Roberts, D. M. (1979). The search for an intelligent evader concealed in one of an arbitrary number of regions. *Naval Res. Logist. Quart.*, 26(4):651–666.

Glazebrook, K., Clarkson, J., and Lin, K. (2019). Fast or slow: search in discrete locations with two search modes. *Oper. Res.*, forthcoming.

Grimmett, G. (1999). *Percolation.* Springer-Verlag, Berlin, second edition.

Hohzaki, R. (2016). Search games: literature and survey. *J. Oper. Res. Soc. Japan*, 59(1):1–34.

Holroyd, A. E., Marcovici, I., and Martin, J. B. (2019). Percolation games, probabilistic cellular automata, and the hard-core model. *Probab. Theory Related Fields*, forthcoming.

Isaacs, R. (1965). *Differential Games. A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* John Wiley & Sons, Inc., New York-London-Sydney.

Jotshi, A. and Batta, R. (2008). Search for an immobile entity on a network. *European J. Oper. Res.*, 191(2):347–359.

Kikuta, K. (1990). A hide and seek game with traveling cost. *J. Oper. Res. Soc. Japan*, 33(2):168–187.

Kikuta, K. (1991). A search game with traveling cost. *J. Oper. Res. Soc. Japan*, 34(4):365–382.

Kikuta, K. (2004). A search game on a cyclic graph. *Naval Res. Logist.*, 51(7):977–993.

Neuts, M. F. (1963). A multistage search game. *J. Soc. Indust. Appl. Math.*, 11:502–507.

Pavlović, L. (1995). A search game on the union of graphs with immobile hider. *Naval Res. Logist.*, 42(8):1177–1189.

Reijnierse, J. H. and Potters, J. A. M. (1993). Search games with immobile hider. *Internat. J. Game Theory*, 21(4):385–394.

Roberts, D. M. and Gittins, J. C. (1978). The search for an intelligent evader; strategies for searcher and evader in the two-region problem. *Naval Res. Logist. Quart.*, 25(1):95–106.

Sakaguchi, M. (1973). Two-sided search games. *J. Operations Res. Soc. Japan*, 16:207–225.

Sorin, S. (2002). *A First Course on Zero-Sum Repeated Games.* Springer-Verlag, Berlin.

Subelman, E. J. (1981). A hide-search game. *J. Appl. Probab.*, 18(3):628–640.

van der Hofstad, R. (2017). *Random Graphs and Complex Networks. Vol. 1.* Cambridge University Press, Cambridge.

von Neumann, J. (1953). A certain zero-sum two-person game equivalent to the optimal assignment problem. In *Contributions to the Theory of Games, vol. 2*, Annals of Mathematics Studies, no. 28, pages 5–12. Princeton University Press, Princeton, N. J.

von Stengel, B. and Werchner, R. (1997). Complexity of searching an immobile hider in a graph. *Discrete Appl. Math.*, 78(1-3):235–249.