

Article

A Multi-Objective Optimization Problem on Evacuating 2 Robots from the Disk in the Face-to-Face Model; Trade-Offs between Worst-Case and Average-Case Analysis [†]

Huda Chuangpishit, Konstantinos Georgiou * and Preeti Sharma

Department of Mathematics, Ryerson University, 350 Victoria St., Toronto, ON M5B 2K3, Canada; h.chuang@ryerson.ca (H.C.); preeti.sharma@ryerson.ca (P.S.)

* Correspondence: konstantinos@ryerson.ca

[†] This paper is an extended version of our paper published in An extended abstract of this work appeared in the proceedings of the 14th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS'18), Helsinki, Finland, 23–24 August 2018.

Received: 7 September 2020; Accepted: 26 October 2020; Published: 29 October 2020



Abstract: The problem of evacuating two robots from the disk in the face-to-face model was first introduced by Czyzowicz et al. [DISC'2014], and has been extensively studied (along with many variations) ever since with respect to worst-case analysis. We initiate the study of the same problem with respect to average-case analysis, which is also equivalent to designing randomized algorithms for the problem. In particular, we introduce constrained optimization problem ${}_2\text{EVAC}_{F2F}$, in which one is trying to minimize the average-case cost of the evacuation algorithm given that the worst-case cost does not exceed w . The problem is of special interest with respect to practical applications, since a common objective in search-and-rescue operations is to minimize the average completion time, given that a certain worst-case threshold is not exceeded, e.g., for safety or limited energy reasons. Our main contribution is the design and analysis of families of new evacuation parameterized algorithms which can solve ${}_2\text{EVAC}_{F2F}$, for every w for which the problem is feasible. Notably, the worst-case analysis of the problem, since its introduction, has been relying on technical numerical, computer-assisted calculations, following tedious robot trajectory analysis. Part of our contribution is a novel systematic procedure, which given *any evacuation algorithm*, can derive its worst- and average-case performance in a clean and unified way.

Keywords: evacuation; disk; face-to-face model; average-case analysis

1. Introduction

In search-type problems, several searchers (robots or mobile agents) are moving within a domain with the objective to identify the location of a (hidden) item. Several variations have been introduced for the problem that range among others with respect to the domain, to searcher specs, to the termination criterion (definition of feasibility) and to the objective of the underlying optimization problem. When in particular there are more than one searchers, and the termination criterion is that *all* searchers reach the hidden item (also known as exit), then the search problem is usually referred to as an *evacuation-type problem*. The term was introduced recently by Czyzowicz et al. in [1] who studied the problem of locating a hidden item on the perimeter of the unit disk with at least two robots. Once all searchers reach the hidden item (hence the name of the problem), the cost of the solution was defined as the time that the last searcher reaches the item (The termination criterion should not be confused with how the cost is quantified.). The problem was studied extensively also by subsequent

publications exclusively under the lens of worst-case (competitive) performance, i.e., with the objective to minimize the worst-case (over all placements of the hidden item) time that the last searcher reaches the exit. In contrast, real-life applications are usually concerned with minimizing the average-case performance of an algorithm, especially if one must solve several instances of the problem. In our case, evacuation-type problems resemble search-and-rescue operations where on one hand it is desirable to minimize the average rescue time, but on the other hand one has to set hard thresholds regarding the possible worst-case performance of the operations (for example, due to safety concerns of the missing person/hidden item).

As a result, we are motivated to revisit existing evacuation-type problems in the realm of average-case performance, where the feasibility criterion is altered so as to also comply with possible bounds on the worst-case performance. Alternatively, one may think of an evacuation-type problem (or even search-type problem) as the multi-objective problem of minimizing simultaneously the worst-case and the average-case cost of feasible evacuation algorithm. In this direction, we study the evacuation problem [1] under the lens of average-case analysis with hard constraints on the worst-case performance, i.e., we study average-case worst-case trade-offs for the underlying optimization problem. More specifically, we introduce new evacuation-type problem ${}_2\text{EVAC}_{F2F}^w$ with two searchers, in which the objective is to optimize the average-case evacuation time (or equivalently to minimize the performance of a randomized algorithm with unbounded access to randomness) condition on that the worst-case evacuation time does not exceed value w . The latter optimization problem seems to be particularly challenging in light of all previous results for the problem that relied on tedious calculations and/or on arguments tailored to the proposed algorithmic solutions. In that direction, we rely on a simple, novel and systematic theoretical approach that allows us to analyze the performance (both average-case and worst-case) of any evacuation algorithm based on computer-assisted calculations. Taking advantage of this approach, we introduce new families of algorithms for ${}_2\text{EVAC}_{F2F}^w$. Interestingly, their worst-case analysis can be done rigorously, giving this way feasible evacuation protocols for the entire spectrum of values w . To motivate our results further, we also investigate the performance of existing algorithms of ${}_2\text{EVAC}_{F2F}^w$, and we actually show that our new algorithms indeed outperform them for many values of w .

1.1. Related Work

In search problems, robots (or commonly referred to as mobile agents or searchers) are equipped with the task of efficiently locating a hidden item in a geometric domain. Studied in many variations since the 1960s, see seminal results [2,3], search problems first focused on identifying optimal probabilistic search and hiding strategies. Several subsequent studies of search-variations gave rise to numerous publications, and eventually to surveys, e.g., [4,5], and books [6–9] that also coined the umbrella term Search Theory. Over the years, search problems were also studied under the lens (i) of exploration by single [10–13] or multiple robots [14–16], (ii) of terrain mapping [17–19] and (iii) of hide-and-peek and pursuit-evasion objectives, e.g., see [20–23].

Although the first reference to a theoretical evacuation problem seems to date back to [24,25], the problem we study follows the line of research of Czyzowicz et al. [1]. Among the many results reported in [1], the one relevant to the current paper pertains to searching with two unit-speed robots for an item/exit hidden at the perimeter of a unit circle. The two searchers operate in the so-called face-to-face model that does not allow exchange of information from distance, and their goal is to minimize the worst-case evacuation time, i.e., the worst-case (over all placements of the exit) time that the last searcher reaches the exit. The first reported upper bound of 5.73906 [1] used a basic evacuation algorithm, according to which robots choose an arbitrary point on the circle, search in opposite directions, while the exit finder detours once the exit is found in order to catch and notify her peer before the evacuate together from the exit. By analyzing the worst placement of the exit [26], improved the upper bound to 5.628 by introducing a carefully chosen detour, before even the exit was reported, that was meant to expedite the catching phase in case the exit was found in a critical

time window. The detour trajectory was later simplified and coupled with an elegant performance analysis that resulted in a further improvement of 5.625 [27]. Only recently, [28] reported yet another improvement of 5.6234, which was achieved by employing multiple detours. In contrast, the best lower bound known of 5.255 is due to [26], and no improvement has been reported since.

Since the inception of ${}_2\text{EVAC}_{F2F}$ in [1], numerous search and evacuation-type variants have emerged that studied different robot specs and/or number of searchers, different communication models, and different domains. Some notable examples include search and/or evacuation in the disk with more than 1 exits [29,30], in triangles [31,32], on multiple rays [33], in graphs [34,35], on a line with at least two robots [36,37] (generalizing the seminal result of [38]), with faulty robots [39–41] or with probabilistically faulty robots [42], with advice (information) [43], with priority specification on the searchers [44,45], with immobile agents [46,47], with time/energy trade-off requirements [48,49], with speed bounds [50], and with terrain dependent speeds [51], just to name some. The interested reader may also see the recent survey [52] that elaborates more on some selected topics.

In a different direction, Baeza-Yates et al. posed the question of minimizing the worst-case trajectory of a single robot searching for a target point at an unknown location in the plane [38]. This was generalized to multiple robots in [53], and later has been studied in [54,55]. More recently, Refs. [56,57] gave lower bounds for related problems. However, in these papers, the robots cannot communicate, and moreover, the objective is for the first robot to find the target.

1.2. Discussion on Closely Related Literature and Improvements

The problem of evacuating robots from the disc was first introduced in [1]. Among the many results reported in the latter paper, the one we expand upon is that of evacuating 2 robots in the face-to-face model. The algorithmic analysis was done exclusively in the deterministic model, and hence was worst-case. The reported upper bound of 5.628 was achieved by a simple algorithm, call it \mathcal{A} , that deploys both robots in an arbitrary point on the perimeter of the unit disc, and then makes them search in different directions. When the exit is found, the finder follows the shortest trajectory (straight line) in order to catch the non-finder in her trajectory so that together they return to the exit.

The first improvement of [26] was motivated by the monotonicity of the cost of \mathcal{A} as a function of the time that the exit is found. A careful analysis shows that the cost function is concave, with the maximizer corresponding to the placement of exit that induces the worst-case performance. The weakness of the underlying communication model is that the (potential) non-finder, call it R_1 has no way to know whether her peer, R_2 has found the exit. At the same time, if enough time has passed and if R_1 has not been caught (notified) by R_2 , then R_1 can deduce that her peer might have found the exit in the “dangerous” neighborhood of the placement inducing the worst-case cost, and hence R_2 might be already moving toward R_1 in order to notify her. As a result, R_1 has the incentive to deviate from searching the perimeter and to move toward the interior to expedite the (possible) rendezvous. Such a detour needs to be carefully chosen so that it is long enough to save cost in case the exit was indeed found in the dangerous neighborhood and short enough so that the detour does not add much in the cost in case it did not. Overall, the time that the detour occurs together with the direction and length of the detour give three degrees of freedom with an objective to optimize the cost in two different cases; when the exit is found in the “dangerous neighborhood” and when it is not. Call this (family of) algorithm(s) \mathcal{A}_0 . The search space for choosing the three optimal parameters for \mathcal{A}_0 is indeed enormous, and at a high level [26] considered only a restriction of the previous idea in which robots agree on a forced meeting on the diameter of the circle, effectively reducing the degrees of freedom of the algorithm but simplifying the already technical and theoretical analysis of its worst-case cost, which eventually relied on numerical calculations.

The upper bound was later improved to 5.625 in [27]. The novelty of the latter work pertained to the introduction of an elegant theoretical analysis of algorithms \mathcal{A}_0 , which on one hand avoided the forced meeting and on the other improved upon the bound of [26] by choosing optimally its three

parameters, which were eventually computed by numerical calculations. The authors of [27] further mentioned that their results could potentially be improved by introducing, at a high level, a recursively defined sequence of detour points aiming to address the reduction of the optimal worst-case cost, should the detour points were one less (so the improvement of \mathcal{A} to \mathcal{A}_0 can be thought as the basic step in the inductive argument). Indeed, ref. [28] used exactly that idea and reported a new upper bound of 5.6234.

Quite interestingly, all previous results pertaining to the worst-case analysis relied on arguments that were somehow tailored to the aforementioned algorithms (robot trajectories). In contrast, we provide a framework that allows us to compute, with computer-assisted calculations, the worst-case cost of any evacuation algorithm, as long as robot trajectories are described by unit-speed-inducing parameterized curves. It is the same framework that allows us also to also compute the average-case cost for any such algorithm.

The main improvement upon the previously results of the same problem pertains to minimizing the average-case cost. In one extreme, one can analyze the naive algorithm in which searchers always stay together, inducing very low average-case cost, but very high worst-case cost. In the other extreme, the optimal algorithm \mathcal{A}_0 of [27] has very low worst-case cost (nearly the best known) and, according to our results, relatively high average-case cost. As a result, we are motivated to minimize the average-case cost, subject to the worst-case cost ranges between the cost of the naive algorithm that of the optimal \mathcal{A}_0 (or equivalently to consider the multi-objective optimization problem of minimizing simultaneously the average-case cost and worst-case cost). Quite surprisingly, we verify, using our technical framework, that among the family of algorithms \mathcal{A}_0 , the one inducing the minimum average-case cost is that of \mathcal{A} . It is the latter observation that gives the main motivation for introducing the three new parameterized families of evacuation algorithms that attempt to minimize their average-case cost, inducing worst-case cost that range, continuously, over the two extreme worst-case costs.

1.3. Outline of Our Results and Paper Organization

We introduce and study a multi-objective analog of the well-studied problem of evacuating two robots from the disk, performing in the face-to-face communication mode. More specifically, we introduce evacuation problem ${}_2\text{EVAC}_{F2F}^w$ in which the two searchers need to evacuate in worst-case time no more than w , while still minimizing their average-case performance. One of our main contributions is a systematic method that allows us to perform both worst-case and average-case analysis for any evacuation algorithm that admits an analytic description. We apply our method to verify that previously known algorithms are not efficient (or not feasible) to ${}_2\text{EVAC}_{F2F}^w$, for certain values of w . As a result, we also motivate the introduction of three new families of evacuation algorithms that are feasible to ${}_2\text{EVAC}_{F2F}^w$ for a wide spectrum of values w , and that induce a continuous bound for the efficient (pareto) frontier for the underlying multi-objective optimization problem. For our results we employ a rigorous analysis for the worst-case performance of the new algorithms, and we rely on numerical computer-assisted calculations and on the aforementioned novel systematic method for estimating their average-case performance.

The formal definition of our problem, along with a high-level exposition of our results appears in Section 2.1. In Section 2.2 our cornerstone observation pertaining to a systematic process for computing the average-case (and worst-case) performance of any evacuation algorithm that admits a convenient representation (as described in Section 2.3). Then in Section 3 we analyze two basic (benchmark) algorithms for ${}_2\text{EVAC}_{F2F}^w$. This allows us to motivate our problem for a range of w values that are related to the benchmark algorithms. In the same section, we also show that the families of previously proposed evacuation algorithms fail to be efficient for ${}_2\text{EVAC}_{F2F}^w$. In Section 4 we present new families of evacuation algorithms, which is also one of our main contributions. For these algorithms we give a rigorous worst-case performance analysis in Section 5, and computer-assisted average-case performance analysis in Section 6, based upon our results in Section 2.2. The reader can also find in the

same section a formal quantification of our results for ${}_{2}\text{EVAC}_{F2F}^w$. Lastly, in Section 7 we conclude by discussing our findings and propose some future directions.

2. Preliminaries

2.1. Problem Definition and Main Results

The geometric domain of ${}_{2}\text{EVAC}_{F2F}$ is a unit disk, centered at the origin of the Cartesian plane. Two unit-speed searchers, starting from the origin, can move anywhere in the disk, and aim to identify a hidden object (exit) located at its perimeter (boundary). The object can be identified only by co-located robot, i.e., a robot that passes over it.

The two robots search in parallel, and perform under a centralized setting, i.e., they know each other’s trajectories under the assumption that no exit is found. The underlying communication model is the so-called face-to-face that does not allow robots to exchange messages from distance, rather only when they meet. A *feasible evacuation* algorithm is a pair of trajectories, one for each robot, in which for every placement of the exit, both robots reach it eventually. For a technical reason, but also without loss of generality, we also require that robots will eventually stay at the exit idle. Placements of the exit i.e., instances to our problem, will be identified by $\text{cycle}(x) := (\cos(x), \sin(x))$, where $x \in [0, 2\pi)$. The evacuation time (cost) $\mathcal{C}(x)$ of a feasible evacuation algorithm on instance $\text{cycle}(x)$ is defined as the first time until both robots reach the exit. t .

In this work we are concerned with determining trade-offs between the worst-case and the average-case performance (of uniform placements of the exit) of evacuation algorithms for ${}_{2}\text{EVAC}_{F2F}$. More specifically, we say that an evacuation algorithm \mathcal{A} with evacuation cost $\mathcal{C}(x)$ on instance $\text{cycle}(x)$ is (a, w) -efficient if

$$\begin{aligned} \text{Avg}(\mathcal{A}) &:= \mathbb{E}_{x \in [0, 2\pi)}[\mathcal{C}(x)] \leq a, \\ \text{Wrs}(\mathcal{A}) &:= \sup_{x \in [0, 2\pi)} \{\mathcal{C}(x)\} \leq w. \end{aligned}$$

where the expectation is with respect to the uniform distribution over $[0, 2\pi)$. Special to our problem is that $\text{Avg}(\mathcal{A})$ can also be interpreted as the expected performance of a randomized algorithm based on \mathcal{A} . Indeed, consider an algorithm which first performs a random rotation of the disk around the origin of angle θ , where θ is chosen uniformly at random from $[0, 2\pi)$, and then simulates \mathcal{A} . Please note that theoretically, this step requires infinitely many random bits, but one can simulate with any precision by enough many random bits. This random step is equivalent to choosing a deployment point uniformly at random on the disk. Due to the symmetry of the domain, it is irrelevant where the adversary will place the unique exit, and hence the expected performance of this randomized algorithm equals $\text{Avg}(\mathcal{A})$.

For algorithms $\mathcal{A}(p)$ parameterized by parameter(s) p , the pair $(\text{Avg}(\mathcal{A}(p)), \text{Wrs}(\mathcal{A}(p)))$ will correspond to a subset of \mathbb{R}^2 (and a curve if p is only one parameter) that we will refer to as the *Efficient Frontier*. We also adopt an optimization perspective of the problem, and we introduce the following optimization problem ${}_{2}\text{EVAC}_{F2F}^w$ on parameter w :

$$\begin{aligned} \min \quad & \frac{1}{2\pi} \int_0^{2\pi} \mathcal{C}(x) dx && ({}_{2}\text{EVAC}_{F2F}^w) \\ \text{s.t.} \quad & \mathcal{C}(x) \leq w, \quad \forall x \in [0, 2\pi). \end{aligned}$$

Please note that the problem above is equivalent to the multi-objective optimization problem $\min\{\text{Avg}(\mathcal{A}(p)), \text{Wrs}(\mathcal{A}(p))\}$, and the conversion to the constrained problem above is due to the well-known ϵ -constraint method, e.g., see [58]. At the same time, no optimal lower bound is known for the worst-case cost in the case of unconstrained average-case cost, not to mention that lower bounds for similar problems are notoriously difficult. Consequently, under the current state-of-the-art it seems

particularly challenging to determine the pareto frontier of the multi-objective optimization problem. Nevertheless, our arguments directly imply bounds for the pareto frontier.

As we show later, the values of w that make ${}_2\text{EVAC}_{F2F}^w$ interesting lie between some special values w_1, w_2 . Values w_1, w_2 are associated with benchmark algorithms, $\mathcal{B}_1, \mathcal{B}_2$, where in particular $\text{Wrs}(\mathcal{B}_1) = w_1 \approx 5.739, \text{Avg}(\mathcal{B}_1) = a_1 \approx 5.1172, \text{Wrs}(\mathcal{B}_2) = w_2 \approx 7.283, \text{Avg}(\mathcal{B}_2) = a_2 \approx 4.28319$. As a result, the reader may think of \mathcal{B}_1 as being inefficient in average case and efficient in worst case. Similarly, \mathcal{B}_2 is inefficient in worst case and efficient in average case.

As is the case for many variations of ${}_2\text{EVAC}_{F2F}$, the cost of best solutions known are computed numerically. Our results pertain to upper bounds for a continuous spectrum of parameter w for problem ${}_2\text{EVAC}_{F2F}^w$. In particular we propose families of algorithms \mathcal{A} (over some parameters) so that as their parameters vary, we obtain $\text{Wrs}(\mathcal{A}) = w$ and $\text{Avg}(\mathcal{A}) = g(w)$, for each $w \in [w_1, w_2]$. The curve $(g(w), w)$ summarizing our results is depicted in Figure 1, and it is later quantified in Theorem 7 (see Section 5).

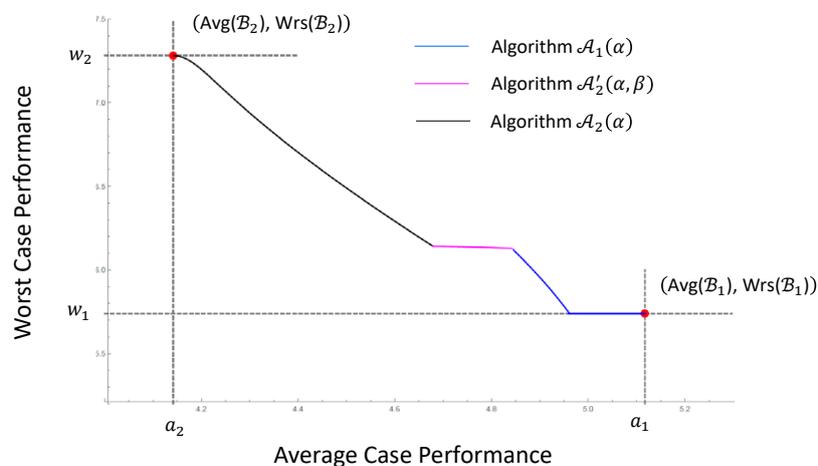


Figure 1. Illustration of the performance of our solution to ${}_2\text{EVAC}_{F2F}^w$, for every $w \in [w_1, w_2]$. Depicted curve corresponds to parametric curve $(g(w), w)$, where $w, g(w)$ are the worst-case performance and average-case performance of three different families of evacuation algorithms $\mathcal{A}_1, \mathcal{A}_2', \mathcal{A}_2$, discussed formally in Section 4. Please note that the magenta curve is not a straight line and, as we show next, induces decreasing worst-case performance (as the average case performance increases).

Please note that an (a, w) -efficient algorithm gives a solution of value a for ${}_2\text{EVAC}_{F2F}^w$. Our approach to prove Theorem 7 is to define families of evacuation algorithms $\mathcal{A}(p)$ parameterized by parameter(s) p . We will prove that these algorithms are $(u(p), v(p))$ -efficient for some functions $u(p), v(p)$, and in particular the evaluation of the worst-case performance will be exact and monotone in p , while the computation of $v(p)$ will be computer-assisted. Then we will set $p = v^{-1}(w)$, and will be able to describe the average-case performance as a function of w as $g(w) := u(v^{-1}(w))$.

2.2. Computing Evacuation Times

For any feasible evacuation algorithm, we denote by $\mathcal{S}(x)$, the first time, after spending time 1 to reach the perimeter that $\text{cycle}(x)$ is visited by any robot. In other words, $\mathcal{S}(x)$ is the time robots spend searching till the exit is found for the first time, assuming that robots do not waste time in the interior of the circle before they start searching. Clearly, when a robot, say \mathcal{R}_1 , locates the exit at $\text{cycle}(x)$, it may attempt to catch \mathcal{R}_2 while moving along \mathcal{R}_2 's trajectory along the shortest line segment, say of length $\mathcal{E}(x)$. Once robots meet, they return together to $\text{cycle}(x)$, inducing total evacuation cost $\mathcal{C}(x) = 1 + \mathcal{S}(x) + 2\mathcal{E}(x)$.

All existing results for ${}_2\text{EVAC}_{F2F}$, from a worst-case complexity perspective, rely on numerical computer-assisted estimation of $\sup_x \mathcal{C}(x)$, after identifying properties of the maximizer. In this section, we elevate existing arguments, and we propose a generalized and unified approach for computing $\mathcal{C}(x)$, for any x and for any robot trajectories. For the sake of formality, as well as for practical purposes, robot trajectories will be defined by parametric functions $\mathcal{F}(t) = (f(t), g(t))$, where $f, g : \mathbb{R} \mapsto \mathbb{R}$ are continuous and piecewise differentiable. In particular, search protocols for the two robots will be given by trajectories $\mathcal{R}_1(t), \mathcal{R}_2(t)$, where $\mathcal{R}_i(t)$ will denote the position of robot \mathcal{R}_i at time $t \geq 0$. Therefore, any evacuation algorithm will be identified by a tuple $(\mathcal{R}_1, \mathcal{R}_2)$. To simplify notation, we will only determine the trajectories from the moment the two robots reach the perimeter of the circle, and until the entire circle is searched (under the assumption that no exit is found), and we will silently assume that robots stay put after exploration is over.

Lemma 1. Consider instance $\text{cycle}(x)$ of ${}_2\text{EVAC}_{F2F}$, and suppose that for a feasible evacuation algorithm $(\mathcal{R}_1, \mathcal{R}_2)$, robot 1 is the first robot that finds the exit. Then $\mathcal{E}(x) = \bar{t} - \mathcal{S}(x)$, where $\bar{t} = \bar{t}(x)$ is the smallest root, no less than $\mathcal{S}(x)$, of function

$$h_x(t) := \|\mathcal{R}_2(t) - \mathcal{R}_1(\mathcal{S}(x))\| - t + \mathcal{S}(x). \tag{1}$$

Proof. The reader may consult Figure 2 that complements our argument.

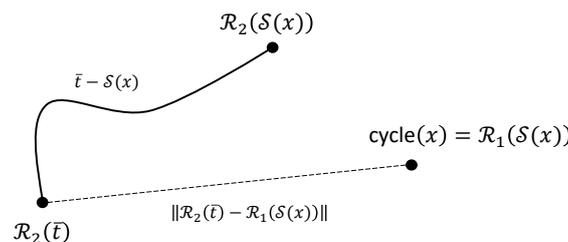


Figure 2. An abstract depiction of the trajectories of $\mathcal{R}_1, \mathcal{R}_2$, assuming that \mathcal{R}_1 is the finder of the exit, located at $\text{cycle}(x)$ at time $\mathcal{S}(x)$. Time \bar{t} is the time, after they start searching for the exit on the perimeter that the two robots meet at $\mathcal{R}_1(\bar{t}) = \mathcal{R}_2(\bar{t})$.

First observe that $h_x(t)$ is continuous. If the robots find the exit together then result holds trivially. So we may assume that that the two robots are not co-located when the exit is found, in which case we have $h_x(\mathcal{S}(x)) > 0$. At the same time, since the evacuation algorithm is feasible, $\mathcal{R}_2(t)$ is eventually a constant, and hence for big enough t we have that $h_x(t)$ becomes eventually negative. By the mean value theorem, there is $t_0 > 0$ for which $h_x(t_0) = 0$.

Now consider the smallest positive root \bar{t} of h_x , no less than $\mathcal{S}(x)$. At time \bar{t} , \mathcal{R}_2 is located at point $\mathcal{R}_2(\bar{t})$, and it is $\|\mathcal{R}_2(\bar{t}) - \mathcal{R}_1(\mathcal{S}(x))\|$ away from the location $\text{cycle}(x)$ of the discovered exit. At the same time, \mathcal{R}_1 moves with speed 1 along the shortest path to catch \mathcal{R}_2 in her trajectory. Hence it takes \mathcal{R}_1 some $\bar{t} - \mathcal{S}(x)$ extra time from the moment the exit is found until she reaches point $\mathcal{R}_2(\bar{t})$. By definition we have $\mathcal{R}_1(\bar{t}) = \mathcal{R}_2(\bar{t})$, and therefore $\mathcal{E}(x) = \bar{t} - \mathcal{S}(x)$ as claimed. \square

For some special trajectories, $\mathcal{E}(x)$ admits a simpler description that we describe next. Before that, we introduce some notation pertaining to a function $\delta : [0, \pi] \mapsto \mathbb{R}_+$, which we widely use in the remaining of the paper:

$$\delta(x) := \text{unique non-negative root (regarding } d) \text{ of " } 2 \sin \left(x + \frac{d}{2} \right) = d \text{ "}. \tag{2}$$

To simplify notation, we will also abbreviate $\delta(x)$ by δ_x . To show that δ_x is well-defined, consider function $f_x(d) = 2 \sin \left(x + \frac{d}{2} \right) - d$. The derivative of the function is $f'_x(d) = \cos x + \frac{d}{2} - 1 \leq 0$.

Since $f'_x(d)$ has also a unique root in $[0, \pi]$, it follows that $f_x(d)$ is strictly decreasing. Observe now that $f_x(0) = 2 \sin(x) \geq 0$, while $f_x(2) = 2 \sin(x + 1) - 2 \leq 0$. We conclude that due to the strict monotonicity of $f_x(d)$, the latter function has indeed unique root in $d \in [0, 2]$. Finally, it is easy to see that $f_x(d) < 0$ when $d > 2$.

Lemma 2. For some instance $\text{cycle}(x)$ of ${}_2\text{EVAC}_{F2F}$, suppose that for a feasible evacuation algorithm $(\mathcal{R}_1, \mathcal{R}_2)$, \mathcal{R}_1 is the finder of the exit, say at time $t_0 = \mathcal{S}(x)$. Assume that both $\mathcal{R}_1(t_0), \mathcal{R}_2(t_0)$ lie on the circle at arc distance 2α , and suppose that \mathcal{R}_2 's movement is along the perimeter of the circle toward the complementary arc of length $2\pi - \alpha$. Then, $\mathcal{E}(x) = \delta_\alpha$.

Proof. The lemma follows by applying transformation $t - \mathcal{S}(x) = d$ in the definition of $h_x(t)$ in Lemma 1, so that $\mathcal{E}(x) = t - \mathcal{S}(x) = d$. \square

We are ready to conclude with a corollary that will be handy for computing evacuation times numerically, and without relying on excessive case analysis, as was the case before.

Corollary 1. Consider feasible evacuation algorithm $(\mathcal{R}_1, \mathcal{R}_2)$ for ${}_2\text{EVAC}_{F2F}$. For any instance $\text{cycle}(x)$ for which \mathcal{R}_1 is the exit finder, the evacuation cost can be computed as $\mathcal{C}(x) = 1 + 2\bar{t} - \mathcal{S}(x)$, where $\bar{t} = \bar{t}(x)$ is the smallest root, at least $\mathcal{S}(x)$, of $h_x(t) := \|\mathcal{R}_2(t) - \mathcal{R}_1(\mathcal{S}(x))\| - t + \mathcal{S}(x)$.

2.3. Trajectory Description

Robot trajectories will be described in phases. We will always omit the “deployment phase”, i.e., the movement from the circle center to its perimeter, and we will only describe the trajectories from the moment robots start searching the circle. In each phase, robot \mathcal{R} , will be moving between two explicit points, either along an arc, or along a line segment (chord of an arc), see Observations 1 and 2 below. We will summarize robot trajectories in tables of the following format.

Robot	Phase #	Trajectory	Duration
\mathcal{R}	1	$\mathcal{R}(t)$	t_1
	2	$\mathcal{R}(t)$	t_2
	\vdots		\vdots

To ease notation, trajectory $\mathcal{R}(t)$ of phase i will be described with parametric equations as if the time is reset to 0 after time $t_0 + t_1 + t_2 + \dots + t_{i-1}$, where $t_0 = 1$ (this is the time that robots reach the circle). The two fundamental trajectory components are movements along arcs and movements along line segments.

Observation 1. Let $b \in [0, 2\pi)$ and $\sigma \in \{-1, 1\}$. The trajectory of an object moving at speed 1 on the perimeter of a unit circle with initial location $\text{cycle}(b)$ is given by the parametric equation $\text{cycle}(\sigma t + b) = (\cos(\sigma t + b), \sin(\sigma t + b))$. If $\sigma = 1$ the movement is counterclockwise (ccw), and clockwise (cw) otherwise.

Proof. It is immediate that when $t = 1$, the object is located at $\text{cycle}(b)$. Its speed is given by calculating $\left\| \frac{\partial}{\partial t} \text{cycle}(\sigma t + b) \right\|$. Indeed, we have

$$\left(\frac{d}{dt} \cos(\sigma t + b) \right)^2 + \left(\frac{d}{dt} \sin(\sigma t + b) \right)^2 = \sigma^2 (-\sin(\sigma t + b))^2 + \sigma^2 (\cos(\sigma t + b))^2 = 1,$$

as wanted. \square

Observation 2. Consider distinct points $A = (a_1, a_2), B = (b_1, b_2)$ in \mathbb{R}^2 . The trajectory of a speed 1 object moving along the line passing through A, B and with initial position A is given by the parametric equation

$$\text{line}(A, B, t) := \left(\frac{b_1 - a_1}{\|A - B\|}t + a_1, \frac{b_2 - a_2}{\|A - B\|}t + a_2 \right).$$

Proof. By definition, the parametric equation above corresponds to a line. Elementary calculations show that $\text{line}(A, B, 0) = A$ and $\text{line}(A, B, \|A - B\|) = B$, that is the object starts at A , and that it passes through B . Object speed is calculated as $\left\| \frac{\partial}{\partial t} \text{line}(A, B, t) \right\|$. In that direction we have

$$\left(\frac{d}{dt} \left(\frac{b_1 - a_1}{\|A - B\|}t + a_1 \right) \right)^2 + \left(\frac{d}{dt} \left(\frac{b_2 - a_2}{\|A - B\|}t + a_2 \right) \right)^2 = \left(\frac{b_1 - a_1}{\|A - B\|} \right)^2 + \left(\frac{b_2 - a_2}{\|A - B\|} \right)^2 = 1$$

as promised. \square

Finally, the analysis of our algorithm trajectories will give rise to several constants. For the reader’s convenience, we list here the numerical values of the most common constants that will be encountered later. $w_1 \approx 5.73906, w_0 \approx 6.11953, w' \approx 6.12851, w_2 \approx 7.28319, \alpha' \approx 1.15468, \bar{\alpha} \approx 1.54419, \beta' \approx 0.0241653, \beta_0 \approx 0.04388$.

All constants are formally defined when they are first introduced.

3. Two Benchmark Algorithms and Motivation

In this section, we describe two benchmark algorithms for ${}_2\text{EVAC}_{F2F}$, as well as perform average-case analysis to algorithms previously proposed in the literature. The reader may consult Figure 3 for the algorithms analyzed in this section.

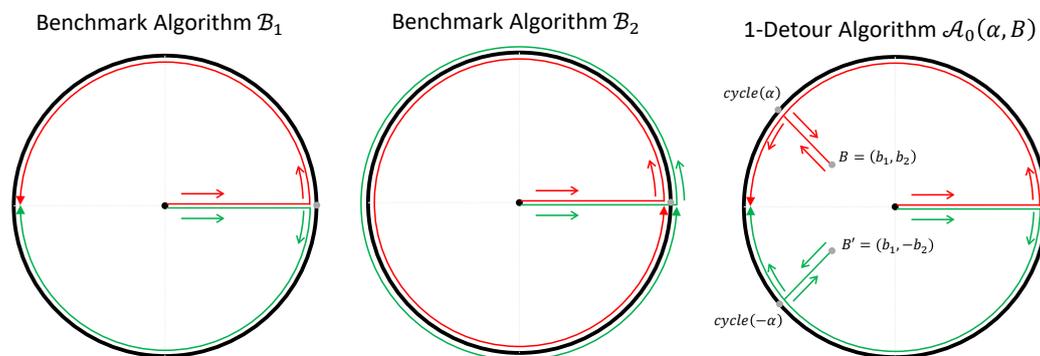


Figure 3. Robot Trajectories for algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{A}_0$. The depicted trajectories show the search of the circle, and not the evacuation step that is performed once the exit is found.

Czyzowicz et al. [1] were the first to introduce an evacuation algorithm for ${}_2\text{EVAC}_{F2F}$, which we denote here by \mathcal{B}_1 (see Figure 3 on the left).

Definition 1 (Benchmark Algorithm \mathcal{B}_1). For all $t \in [0, \pi]$, $\mathcal{R}_1(t) = \text{cycle}(t)$ and $\mathcal{R}_2(t) = \text{cycle}(-t)$.

Observation 3. Benchmark Algorithm \mathcal{B}_1 is $(5.1172, 5.73906)$ -efficient.

Proof. Please note that it takes time π to search the entire circle, and that the two trajectories are symmetric with respect to horizontal axis. Therefore, we may assume that the instance $\text{cycle}(x)$ satisfies $x \in [0, \pi]$.

Clearly, for any such x , we have that $\mathcal{S}(x) = x$. By Lemma 2, we have that $\mathcal{C}(x) = 1 + \mathcal{S}(x) + 2\mathcal{E}(x) = 1 + x + 2\delta_x$. Numerical calculations (software assisted) show that

$$\begin{aligned} \text{Wrs}(\mathcal{B}_1) &= \sup_{x \in [0, \pi]} \{\mathcal{C}(x)\} = \sup_{x \in [0, \pi]} \{1 + x + 2\delta_x\} \approx 5.73906, \\ \text{Avg}(\mathcal{B}_1) &= \mathbb{E}_{x \in [0, \pi]}[\mathcal{C}(x)] = \frac{1}{\pi} \int_{x=0}^{\pi} (1 + x + 2\delta_x) dx \approx 5.1172. \end{aligned}$$

□

\mathcal{B}_1 should be understood as being efficient in the worst case, but inefficient on average. The claim becomes transparent by introducing the following *naive* algorithm for ${}^2\text{EVAC}_{F2F}$ that we depict in the middle of Figure 3.

Definition 2 (Benchmark Algorithm \mathcal{B}_2). For each $t \in [0, 2\pi]$, $\mathcal{B}_1(t) = \mathcal{B}_2(t) = \text{cycle}(t)$.

Observation 4. Benchmark Algorithm \mathcal{B}_2 is $(1 + \pi, 1 + 2\pi)$ -efficient.

Proof. It is easy to see that for all $x \in [0, 2\pi]$ we have $\bar{f}(x) = \mathcal{S}(x) = x$ and $\mathcal{E}(x) = 0$. Therefore $\mathcal{C}(x) = 1 + x$, and hence

$$\begin{aligned} \text{Wrs}(\mathcal{B}_2) &= \sup_{x \in [0, 2\pi]} \{\mathcal{C}(x)\} = 1 + 2\pi, \\ \text{Avg}(\mathcal{B}_2) &= \mathbb{E}_{x \in [0, 2\pi]}[\mathcal{C}(x)] = \int_{x=0}^{2\pi} (1 + x) dx = 1 + \pi. \end{aligned}$$

□

\mathcal{B}_2 should be understood as highly efficient on average, but inefficient in the worst case. Moreover, it should be clear that $\mathcal{B}_1, \mathcal{B}_2$ are feasible solutions to ${}^2\text{EVAC}_{F2F}^w$, for $w = 5.1172$ and $w = 1 + 2\pi$, respectively. We conjecture that \mathcal{B}_1 is indeed the optimal evacuation algorithm among all algorithms with worst-case performance no more than $1 + 2\pi$. At the same time, below we show that \mathcal{B}_2 is the best algorithm for ${}^2\text{EVAC}_{F2F}^w$, when $w = 5.1172$, among those previously used to improve upon the worst-case performance up to the third decimal. The importance of this observation is two-fold; first we are motivated to study ${}^2\text{EVAC}_{F2F}^w$ for the entire spectrum of $w \in [\text{Wrs}(\mathcal{B}_1), \text{Wrs}(\mathcal{B}_2)]$, and second we deduce that in order to perform well on average, we need to devise and analyze new evacuation algorithms.

Upper bounds for the worst-case performance of \mathcal{B}_1 were later improved first to 5.628 [26], then to 5.625 [27], and then to 5.623 [28]. The main idea behind the improvement is to understand the monotonicity of $\mathcal{C}(x)$ for algorithm \mathcal{B}_1 . Indeed, the following lemma was implicit in both [26,27], and can be obtained numerically.

Lemma 3. There is α_0 , where $\alpha_0 \approx 0.96782$, so that evacuation cost $\mathcal{C}(x)$ of \mathcal{B}_1 for ${}^2\text{EVAC}_{F2F}$ on instance $\text{cycle}(x)$ is strictly increasing for $x \in [0, \alpha_0]$, and strictly decreasing in $x \in [\alpha_0, \pi]$. In particular, $\text{Wrs}(\mathcal{B}_1) = \mathcal{C}(\alpha_0) \approx 5.73906$.

Consider now an execution of \mathcal{B}_1 in which one of the robots, say \mathcal{R}_2 continues searching on the circle and is close to approach a location that would be the meeting point if the instance was $\text{cycle}(\alpha_0)$. In an attempt to help expedite a potential meeting (in case \mathcal{R}_1 is approaching) and effectively reducing the cost of the worst case, \mathcal{R}_2 would make a minor detour toward the interior of the disk, before returning back to the exploration of the circle. This simple idea was explored in [26] and in [27] where the following family of algorithms were introduced, parameterized by $\alpha \in [0, \pi]$ and point B within the unit disk, see also right of Figure 3 (the simplified version presented here is due to [27]).

Definition 3 (1-Detour Algorithm $\mathcal{A}_0(\alpha, B)$). For all $t \in [0, \pi + 2 \|\text{cycle}(\alpha) - B\|]$, the trajectory of \mathcal{R}_1 is defined as

Robot	Phase #	Trajectory	Duration
\mathcal{R}_1	1	cycle(t)	α
	2	line(cycle(α), B, t)	$\ \text{cycle}(\alpha) - B\ $
	3	line($B, \text{cycle}(\alpha), t$)	$\ \text{cycle}(\alpha) - B\ $
	4	cycle($t + \alpha$)	$\pi - \alpha$

The trajectory of \mathcal{R}_2 is symmetric with respect to the horizontal axis.

The crux of the contribution of [26] was to prove that there exists α, B for which the worst-case performance is no more than 5.644 (and a delicate refinement is needed to achieve 5.628). Notably, their analysis is tedious and lengthy, whereas we can obtain the same result, relying again on numerical calculations, with minimal effort. Then, [27] proposed variations of $\mathcal{A}_0(\alpha, B)$ in which each robot performs more than 1 detours (see Phases 2,3 of $\mathcal{A}_0(\alpha, B)$), giving rise to [28]. Hence, t -detour algorithms are parameterized by a sequence $\alpha_1, \dots, \alpha_t$, where $\alpha_i \geq 0$ and $\sum_i \alpha_i \leq \pi$, and points B_i in the disk. Even 2-detour algorithms achieve worst-case performance 5.623, while for each $t \geq 2$, t -detour algorithms do induce strictly improved performance (for appropriate choices of the parameters) but the improvement seems to be negligible.

Motivated by the results in [26,27], one is tempted to ask whether any algorithm in the family $\mathcal{A}_0(\alpha, B)$ improves upon \mathcal{B}_1 with respect to the average-case analysis. The next claim is due to exhaustive, computer-assisted numerical calculations, see also Figure 4.

Theorem 5. For every $\alpha \in [0, \pi)$ and for every B in the unit disk $\text{Avg}(\mathcal{A}_0(\alpha, B)) \geq \text{Avg}(\mathcal{B}_1)$.

Theorem 5 provides strong motivation for studying problem ${}_2\text{EVAC}_{F2F}^w$, since it shows that in order to establish good upper bounds, i.e., our main results depicted in Figure 1 and quantified later in Theorem 7, one needs to employ new evacuation algorithms. Recall that even $\text{Wrs}(\mathcal{B}_1)$ and $\text{Wrs}(\mathcal{A}_0(\alpha, B))$ were estimated with computer-assisted calculations. Due to the nature of the problem, we are bound to rely on computer-assisted calculations as well. Notably, our much more intense computational work is feasible only because we employ the new method for computing evacuation times due to Corollary 1 and Definition 3 of $\mathcal{A}_0(\alpha, B)$ trajectories. Overall, to verify Theorem 5 we compute pairs $(\text{Avg}(\mathcal{A}_0(\alpha, B)), \text{Wrs}(\mathcal{A}_0(\alpha, B)))$ for more than 500,000 different parameter values and we depict them in Figure 4.

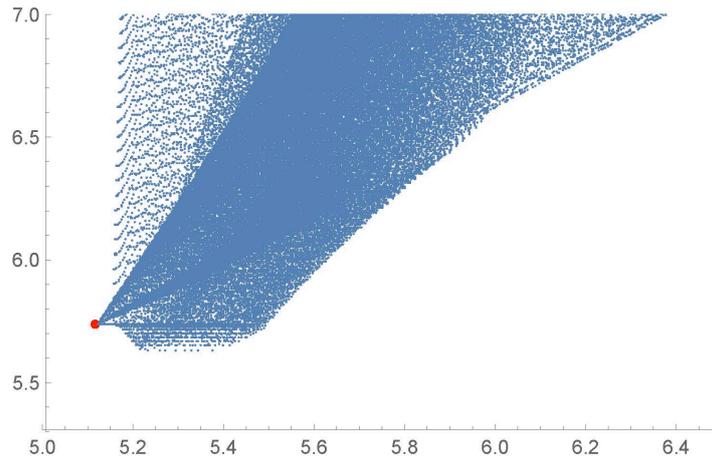


Figure 4. Performance analysis of $\mathcal{A}_0(\alpha, B)$ for various values of parameters α, B . Blue points (a, w) correspond to (a, w) -efficient algorithms $\mathcal{A}_0(\alpha, B)$. The red point is $(\text{Avg}(\mathcal{B}_1), \text{Wrs}(\mathcal{B}_1))$, i.e., the performance of \mathcal{B}_1 in the average-worst-case space. Please note that no algorithm \mathcal{A}_0 performs better on average than \mathcal{B}_1 , while all $\mathcal{A}_0(t, \text{cycle}(t))$ is exactly \mathcal{B}_1 for every point $t \in [0, \pi]$. Notably, all points lie above the threshold of worst-case performance 5.625, and some are arbitrarily close to that value (corresponding to choices of α, B that give the algorithm of [27]).

4. New Evacuation Algorithms

In this section, we propose families of evacuation algorithms for problem ${}^2\text{EVAC}_{F2F}^w$, for the entire spectrum of $w \in [\text{Wrs}(\mathcal{B}_1), \text{Wrs}(\mathcal{B}_2)]$. Our algorithms are summarized in Figure 5.

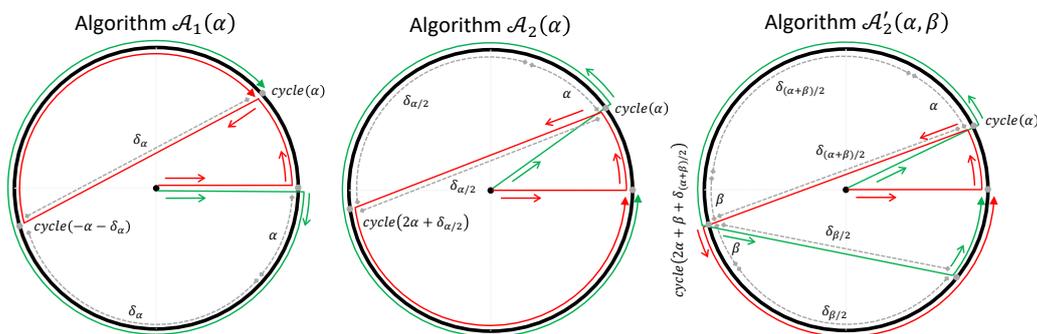


Figure 5. Robot Trajectories for algorithms $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2$. The depicted trajectories show the search of the circle, and not the evacuation step that is performed once the exit is found. Arcs that are searched by both robots are also searched simultaneously, i.e., robots are co-located and search together.

First we define families of evacuation algorithms that, as we show next, perform well for ${}^2\text{EVAC}_{F2F}^w$ in the “neighborhood of \mathcal{B}_1 ”, i.e., for w close to $\text{Wrs}(\mathcal{B}_1)$. Our algorithms are parameterized by α , and their circle exploration lasts $2\pi - \alpha$.

Definition 4 (Algorithm $\mathcal{A}_1(\alpha)$). For all $t \in [0, 2\pi - \alpha]$, the trajectory of \mathcal{R}_1 is defined as

Robot	Phase #	Trajectory	Duration
\mathcal{R}_1	1	$\text{cycle}(t)$	α
	2	$\text{line}(\text{cycle}(\alpha), \text{cycle}(-\alpha - \delta_\alpha), t)$	δ_α
	3	$\text{cycle}(-\alpha - \delta_\alpha - t)$	$2\pi - 2\alpha - \delta_\alpha$

where δ_a is defined in (2). The trajectory of \mathcal{R}_2 is defined as $\mathcal{B}_2(t) = \text{cycle}(-t)$, for all $t \in [0, 2\pi - \alpha]$.

\mathcal{A}_1 is depicted in Figure 5 on the left. At a high level $\mathcal{A}_1(\alpha)$ is a modification of \mathcal{B}_1 that is based on the following idea. The execution of $\mathcal{A}_1(\alpha)$ is the same as in \mathcal{B}_1 until each robot searches an arc

length α (and hence $\mathcal{A}(\pi)$ coincides with \mathcal{B}_1). After time α , \mathcal{R}_1 abandons her trajectory and catches \mathcal{R}_2 , on the perimeter of the circle resembling a trajectory as if the exit was located at $\mathcal{R}_1(\alpha)$. It is not difficult to see that the definition of δ_α above satisfies $\mathcal{R}_1(\alpha + \delta_\alpha) = \mathcal{R}_2(\alpha + \delta_\alpha) = \text{cycle}(-\alpha - \delta_\alpha)$.

Next we define a family of algorithms \mathcal{A}_2 which, as we show later, perform well in the “neighborhood of \mathcal{B}_2 ”, i.e., for w close to $\text{Wrs}(\mathcal{B}_2)$. For this recall definition (2) of δ_a . We let $\gamma_0 \approx 2.2412$ be the root of $2\alpha + \delta_{\alpha/2} = 2\pi$. For every $\alpha \leq \gamma_0$ we define a family of algorithms on parameter α whose circle exploration lasts $2\pi - \alpha$.

Definition 5 (Algorithm $\mathcal{A}_2(\alpha)$). For all $t \in [0, 2\pi - \alpha]$, the trajectory of \mathcal{R}_1 is defined as

Robot	Phase #	Trajectory	Duration
\mathcal{R}_1	1	cycle(t)	α
	2	line(cycle(α), cycle($2\alpha + \delta_{\alpha/2}$), t)	$\delta_{\alpha/2}$
	3	cycle($2\alpha + \delta_{\alpha/2} + t$)	$2\pi - 2\alpha - \delta_{\alpha/2}$

The trajectory of \mathcal{R}_2 is defined as $\mathcal{R}_2(t) = \text{cycle}(\alpha + t)$, for all $t \in [0, 2\pi - \alpha]$.

\mathcal{A}_2 is depicted in the middle of Figure 5. The condition that $\alpha \leq \gamma_0$ is added for simplicity to ensure that the latest catching point occurs while the other robot is still searching, and is not mandatory. At a high level $\mathcal{A}_2(\alpha)$ is a generalization of \mathcal{B}_2 (note that $\mathcal{A}_2(0) = \mathcal{B}_2$). For the first α time units, robots search in the same direction until \mathcal{R}_1 arrives at the deployment point of \mathcal{R}_2 . Then, \mathcal{R}_1 catches \mathcal{R}_2 on the circle, as if the exit was located at $\mathcal{R}_1(\alpha)$ (which by Lemma 2 happens in $\delta_{\alpha/2}$ extra time).

Finally, we introduce a family of evacuation algorithms which will perform well for ${}_2\text{EVAC}_{F2F}^w$ for intermediate values of $w \in [\text{Wrs}(\mathcal{B}_1), \text{Wrs}(\mathcal{B}_2)]$. For this we generalize family \mathcal{A}_2 so that the two robots perform two alternating jumps, with parameters α, β satisfying $2\alpha + 2\beta + \delta_{(\alpha+\beta)/2} + \delta_{\beta/2} \leq 2\pi$, see right of Figure 5.

Definition 6 (Algorithm $\mathcal{A}'_2(\alpha, \beta)$). For notational convenience, we set $\zeta_{\alpha,\beta} := 2\alpha + \beta + \delta_{(\alpha+\beta)/2}$. For all $t \in [0, 2\pi - \alpha - \beta]$, the trajectories of $\mathcal{R}_1, \mathcal{R}_2$ are defined as follows

Robot	Phase #	Trajectory	Duration
\mathcal{R}_1	1	cycle(t)	α
	2	line(cycle(α), cycle($\zeta_{\alpha,\beta}$), t)	$\delta_{(\alpha+\beta)/2}$
	3	cycle($\zeta_{\alpha,\beta} + t$)	$2\pi - 2\alpha - \beta - \delta_{(\alpha+\beta)/2}$
\mathcal{R}_2	1	cycle($\alpha + t$)	$\alpha + \beta + \delta_{(\alpha+\beta)/2}$
	2	line(cycle($\zeta_{\alpha,\beta}$), cycle($\zeta_{\alpha,\beta} + \delta_{\beta/2}$), t)	$\delta_{\beta/2}$
	3	cycle($\zeta_{\alpha,\beta} + \beta + \delta_{\beta/2} + t$)	$2\pi - 2\alpha - 2\beta - \delta_{(\alpha+\beta)/2} - \delta_{\beta/2}$

Next we describe the meaning of parameters α, β of the Robot trajectories above. As in the family of algorithms \mathcal{A}_2 , parameter α represents the arc distance the two robots have before the one preceding decides to jump ahead. In \mathcal{A}_2 the two robots meet again once the jumper reaches the perimeter of the circle. In \mathcal{A}'_2 the jumper deploys a little further away on the circle so that when the other robot reaches the deployment point of the jumper, the two robots are at arc distance β . As a result, the time it takes both robots to complete searching the entire circle is $2\pi - \alpha - \beta$, as well as $\mathcal{A}'_2(\alpha, 0)$ coincides with $\mathcal{A}_2(\alpha)$. Finally, note that even though \mathcal{A}'_2 will be eventually invoked for seemingly restricted values of β ($\beta \leq \beta_0 \approx 0.04388$), the deviation in the performance will be significant enough (e.g., $\delta_{\beta_0/2} \approx 0.977997$) to account for its use in our upper bounds.

5. Worst-Case Performance Analysis

In this section, we perform worst-case analysis for all algorithmic families $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2$ with respect to their parameters. Notably, results in this section are quantified formally and exactly by closed

formulas. At a high level, each of $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2$ will be invoked to solve ${}_{2}\text{EVAC}_{F2F}^w$ for different values of $w \in [\text{Wrs}(\mathcal{B}_1), \text{Wrs}(\mathcal{B}_2)]$, and each of them will have competitive average-case performance for the corresponding worst-case performance w . As an easy warm-up, we analyze \mathcal{A}_1 .

Lemma 4 (Worst-Case Analysis for \mathcal{A}_1). *Let $\bar{\alpha} = 1 + 2\pi - w_1$, where $w_1 = \text{Wrs}(\mathcal{B}_1)$. Then, for all $\alpha \in [0, \pi]$, we have that*

$$\text{Wrs}(\mathcal{A}_1(\alpha)) = \begin{cases} 1 + 2\pi - \alpha & , \forall \alpha \in [0, \bar{\alpha}) \\ \text{Wrs}(\mathcal{B}_1) & , \forall \alpha \in [\bar{\alpha}, \pi] \end{cases} .$$

Proof. First it is easy to show that the worst-case evacuation time is induced either when \mathcal{R}_1 finds the exit while moving from cycle(0) to cycle(α), or while $\mathcal{R}_1, \mathcal{R}_2$ are exploring the circle together (after having met). By Lemma 2, the cost in the first case would be

$$\max_{0 \leq x \leq \alpha} \{1 + x + 2\delta_x\} = \begin{cases} 1 + \alpha + 2\delta_\alpha & , \text{if } \alpha \leq \alpha_0 \\ \text{Wrs}(\mathcal{B}_1) & , \text{otherwise} \end{cases}$$

where the values of the piecewise function above follow from Lemma 3. In the other case, the worst placement of exit is obtained using instances cycle($\alpha + \epsilon$) for arbitrary small values of $\epsilon > 0$ in which case the evacuation cost becomes $1 + 2\pi - \alpha$.

Overall, it is easy to see that $1 + \alpha_0 + 2\delta_{\alpha_0} \leq 1 + 2\pi - \alpha_0$ showing that the dominant evacuation cost when $\alpha \leq \bar{\alpha}$ is $1 + 2\pi - \alpha$. For $\alpha > \bar{\alpha}$ the evacuation cost becomes equal to w_1 . \square

In a similar fashion, we can easily analyze \mathcal{A}_2 .

Lemma 5 (Worst-Case Analysis for \mathcal{A}_2). *For all $\alpha \leq \pi - 2$, we have $\text{Wrs}(\mathcal{A}_2(\alpha)) = 1 + 2\pi - \alpha$.*

Proof. We distinguish three cases as to where the exit is. If $x \in [0, \alpha)$, then the worst instance cycle(x) is when $x = \alpha - \epsilon$ for arbitrarily small $\epsilon > 0$, and the cost is $1 + \alpha + 2\delta_{\alpha/2}$. In the second case $x \in [\alpha, 2\alpha + \delta_{\alpha/2})$ and it is not difficult to see that the worst-case induced cost in this case is not more than that of the first case. Finally, in the third case $x \in [2\alpha + \delta_{\alpha/2}, 2\pi)$, and the two robots move together, so the total cost, in the worst-case, is $1 + 2\pi - \alpha$, when $x = 2\pi - \epsilon$ for arbitrarily small $\epsilon > 0$. It is not difficult to see that the dominant case is actually the third one, and in fact the two cases induce the same cost when $\pi = \alpha + \delta_{\alpha/2}$. By the definition of $\delta_{\alpha/2}$ we know that $\delta_{\alpha/2} = 2 \sin\left(\frac{\alpha + \delta_{\alpha/2}}{2}\right) = 2 \sin(\pi/2) = 2$. Hence, the costs become equal when $\alpha = \pi - 2$. \square

Next, we analyze $\mathcal{A}'_2(\alpha, \beta)$, which requires more technical arguments. For this we will invoke \mathcal{A}'_2 only for special parameters, whose choice is motivated by the following observation pertaining to the performance of \mathcal{A}_2 (whose generalization is \mathcal{A}'_2). From the proof of Lemma 5, it follows that among all algorithms $\mathcal{A}_2(\alpha)$, where $\alpha \leq \gamma_0$ (see discussion before Definition 5), the one with minimum worst-case evacuation cost is $\mathcal{A}_2(\pi - 2)$, and the cost becomes $3 + \pi$. In fact, for all $w \in [3 + \pi, 1 + 2\pi]$ there are two different values of α for which $\text{Wrs}(\mathcal{A}_2(\alpha)) = w$, and we restrict $\alpha \in [0, \pi - 2]$ so that we obtain evacuation algorithms with minimum average-case cost. Moreover, $\alpha = \pi - 2$ is the only parameter for which $\text{Wrs}(\mathcal{A}_2(\alpha)) = 3 + \pi$ and as a byproduct, it is the algorithm in the family \mathcal{A}_2 that minimizes the worst-case.

By Lemma 5 we know that as $\beta \rightarrow 0$, the value of α that minimizes $\text{Wrs}(\mathcal{A}'_2(\alpha, \beta))$ approaches $\pi - 2$. That value of α is what made the evacuation cost of $\mathcal{A}_2(\alpha)$ attain the same value in two different (worst-case) exit placements. Motivated by this, and for values of $\beta > 0$ not too big, we still find the optimal choices of α that minimize the worst-case performance.

Lemma 6 (Worst-Case Analysis for \mathcal{A}'_2). *Let $\beta_0 = 0.0438855$, and set $\alpha_\beta := \pi - \beta/2 - 2 \cos(\beta/4)$. Then for all $\beta \in [0, \beta_0]$ we have $\text{Wrs}(\mathcal{A}'_2(\alpha_\beta, \beta)) = 1 + \pi - \beta/2 + 2 \cos(\beta/4)$.*

Proof. Let $w(\beta) = 1 + \pi - \beta/2 + 2 \cos(\beta/4)$. First we show that $w(\beta)$ is the worst-case performance of $\mathcal{A}'_2(\alpha_\beta, \beta)$ for two specific placements of the exit.

We proceed by describing evacuation cost $\mathcal{C}(x)$ assuming two arbitrary α, β for two different instances $\text{cycle}(x)$. Using Lemma 2, we see that

$$\lim_{\epsilon \rightarrow 0^+} \mathcal{C}(\alpha - \epsilon) = 1 + \lim_{\epsilon \rightarrow 0^+} \mathcal{S}(\alpha - \epsilon) + 2 \lim_{\epsilon \rightarrow 0^+} \mathcal{E}(\alpha - \epsilon) = 1 + \alpha + 2\delta_{\alpha/2}. \tag{3}$$

Since the total search time is $2\pi - \alpha - \beta$, we also see that

$$\lim_{\epsilon \rightarrow 0^+} \mathcal{C}(2\pi - \epsilon) = 1 + 2\pi - \alpha - \beta. \tag{4}$$

Now we claim that (3), (4) are equal when $\alpha = \alpha_\beta$. Indeed, equating (3), (4) gives

$$a + \delta_{\alpha/2} = \pi - \beta/2. \tag{5}$$

However, then, using (2), we see that

$$\delta_{\alpha/2} = 2 \sin\left(\frac{\alpha + \delta_{\alpha/2}}{2}\right) = 2 \sin\left(\frac{\pi - \beta/2}{2}\right) = 2 \cos(\beta/4). \tag{6}$$

Substituting (6) into (5), we see that the value of α for which (3), (4) are equal satisfies $\alpha = \pi - \beta/2 - 2 \cos(\beta/4)$, as promised. Substituting this special value of $\alpha = \alpha_\beta$ either in (3) or in (4) induces evacuation cost $w(\beta) = 1 + \pi - \beta/2 + 2 \cos(\beta/4)$.

Next we show that as long as β is not too big, $w(\beta)$ is indeed the worst-case evacuation cost. We consider the following cases $x \in I_i, i = 1, \dots, 4$ for possible instances $\text{cycle}(x)$;

$$\begin{aligned} I_1 &:= [0, \alpha), \\ I_2 &:= [\alpha, 2\alpha + \beta + \delta_{(\alpha+\beta)/2}), \\ I_3 &:= [2\alpha + \beta + \delta_{(\alpha+\beta)/2}, 2\alpha + 2\beta + \delta_{(\alpha+\beta)/2} + \delta_{\beta/2}), \\ I_4 &:= [2\alpha + 2\beta + \delta_{(\alpha+\beta)/2} + \delta_{\beta/2}, 2\pi). \end{aligned}$$

Clearly, (3), (4) demonstrate the worst-case evacuation costs for instances in I_1, I_4 , respectively, and the cost in both cases, for $\alpha = \alpha_\beta$ is equal to $w(\beta)$.

If $x \in I_2$ then $\mathcal{C}(x) = 1 + \mathcal{S}(x) + 2\mathcal{E}(x)$. It is easy to see that both $\mathcal{S}(x), \mathcal{E}(x)$ are monotone in I_2 , so the worst-case evacuation in this case is

$$\lim_{\epsilon \rightarrow 0^+} \mathcal{C}(2\alpha_\beta + \beta + \delta_{(\alpha_\beta+\beta)/2} - \epsilon) = 1 + \alpha_\beta + \beta + \delta_{(\alpha_\beta+\beta)/2} + 2\delta_{\beta/2}. \tag{7}$$

Denote $\delta_{\beta/2}$ satisfying (2) by δ'_β . Using (2) and the definition of α_β , we see that

$$\delta_{(\alpha_\beta+\beta)/2} = 2 \sin\left(\frac{\alpha_\beta + \beta + \delta_{(\alpha_\beta+\beta)/2}}{2}\right) = 2 \cos\left(\cos(\beta/4) - \beta/4 - \delta_{(\alpha_\beta+\beta)/2}\right).$$

For simplicity, we denote $\delta_{(\alpha_\beta+\beta)/2}$ that satisfies the equation above by δ''_β . Then, continuing from (7), the worst-case evacuation cost when $x \in I_2$ becomes $1 + \pi + \beta/2 - 2 \cos(\beta/4) + \delta''_\beta + 2\delta'_\beta$, an expression that depends exclusively on β . The latter cost is no more than $w(\beta)$ if and only if $4 \cos(\beta/4) - \beta - \delta''_\beta - 2\delta'_\beta \geq 0$, and numerically we verify that this is satisfied as long as $\beta \leq \beta_0$ (see also Figure 6).

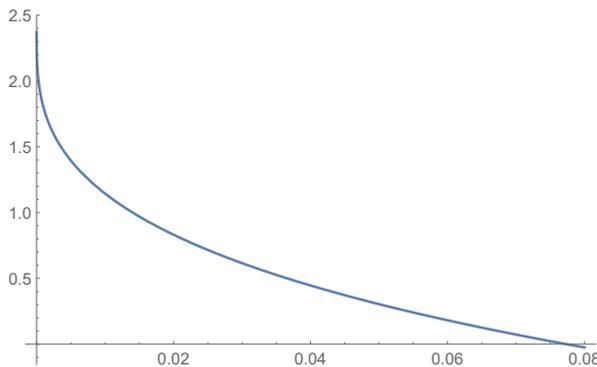


Figure 6. The behavior of expression $4 \cos(\beta/4) - \beta - \delta''_{\beta} - 2\delta'_{\beta}$, for $\beta = 0, \dots, 0.8$.

Finally, it is easy to verify that $\delta_{\beta/2}$ and $|I_4|$ are increasing and decreasing, respectively, for $\beta \leq \beta_0$, and that $\delta_{\beta_0/2} = 0.977997 \leq 1.01099 = |I_4|$ (for $\beta = \beta_0$). As a result, the worst-case evacuation cost of case $x \in I_3$ cannot exceed that of case $x \in I_4$, and hence the lemma follows. \square

It is important to note that the worst-case performance of $\mathcal{A}'_2(\alpha_{\beta}, \beta)$ of Lemma 7 is decreasing in β . Indeed, $\frac{\partial}{\partial \beta} \text{Wrs}(\mathcal{A}'_2(\alpha_{\beta}, \beta)) = \frac{1}{2} - \frac{1}{2} \sin(\beta/4) < 0$. For values of β close to 0, the derivative is nearly a constant. This also explains why in Figure 1, the performance of algorithm \mathcal{A}'_2 seems to have nearly invariant worst-case performance, which however is provably decreasing in β .

Lemma 7 (Worst-Case Analysis for \mathcal{A}'_2). Let $\beta_0 = 0.0438855$, and set $\alpha_{\beta} := \pi - \beta/2 - 2 \cos(\beta/4)$. Then for all $\beta \in [0, \beta_0]$ we have $\text{Wrs}(\mathcal{A}'_2(\alpha_{\beta}, \beta)) = 1 + \pi - \beta/2 + 2 \cos(\beta/4)$.

6. Average-Case Performance Analysis and the Efficient Frontier

In this section, we perform average-case analysis for all algorithmic families $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2$, with respect to their parameters. For the sake of exposition of our results, we set $w_1 = \text{Wrs}(\mathcal{B}_1) \approx 5.73906, w_2 = \text{Wrs}(\mathcal{B}_2) = 1 + 2\pi \approx 7.28319$ and for $\beta_0 \approx 0.04388$, as in Lemma 7, we set $w_0 := \text{Wrs}(\mathcal{A}'_2(\alpha_{\beta_0}, \beta_0)) \approx 6.11953$. We also recall $\bar{\alpha} \approx 1.54419$ of Lemma 4. Finally, we set

$$\begin{aligned} v(\alpha) &:= 1 + 2\pi - \alpha \\ v_2(\beta) &:= 1 + \pi - \beta/2 + 2 \cos(\beta/4) \\ u_1(\alpha) &:= 0.00889\alpha^3 - 0.16944\alpha^2 + 0.71518\alpha + 4.23089 \\ u'_2(\beta) &:= 530.673\beta^3 - 78.5498\beta^2 + 7.36219\beta + 4.70493 \\ u_2(\alpha) &:= 0.093056\alpha^2 + 0.346659\alpha + 4.1719 \end{aligned}$$

Combined with our findings of Section 5, the main result of the current section is the following.

Theorem 6. For every $w \in [w_1, w_2]$ there is algorithm $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}'_2, \mathcal{A}_2\}$ and unique parameter(s) p such that $\text{Wrs}(\mathcal{A}(p)) = w$. In particular,

- for all $\alpha \in [1, \bar{\alpha}]$, $\mathcal{A}_1(\alpha)$ is $(u_1(\alpha), v(\alpha))$ -efficient, and $v([1, \bar{\alpha}]) = [w_1, 2\pi]$,
- for all $\beta \in [0, \beta_0]$, $\mathcal{A}'_2(\alpha_{\beta}, \beta)$ is $(u'_2(\beta), v_2(\beta))$ -efficient, and $v_2([0, \beta_0]) = [w_0, 3 + \pi]$,
- for all $\alpha \in [0, \pi - 2]$, $\mathcal{A}_2(\alpha)$ is $(u_2(\alpha), v(\alpha))$ -efficient, and $v([0, \pi - 2]) = [3 + \pi, w_2]$.

Proof. The claims for the worst-case performances of $\mathcal{A}_1, \mathcal{A}'_2, \mathcal{A}_2$ follow directly from Lemmata 4, 7 and 5, respectively. Next we argue that as the parameters vary in their specified range, we obtain the entire spectrum of $w \in [w_1, w_2]$, and this for unique values of the parameters. For this, we will rely on that for all evacuation algorithm families, the worst-case cost is monotone in the parameters.

First, we argue about \mathcal{A}_1 . We observe that by the definition of $\bar{\alpha}$, $\text{Wrs}(\mathcal{A}_1(\bar{\alpha})) = w_1$, and $\text{Wrs}(\mathcal{A}_1(1)) = 1 + 2\pi - 1 = 2\pi$. Together with the fact that $v(\alpha)$ is strictly decreasing, we see that $\text{Wrs}(\mathcal{A}_1(\alpha))$ is 1-1 and onto to $[w_1, 2\pi]$ as α ranges in $[1, \bar{\alpha}]$.

Second, we study \mathcal{A}'_2 whose worst-case cost $v_2(\beta)$ is strictly decreasing in β . Moreover, by definition of β_0 , we have $\text{Wrs}(\mathcal{A}'_2(\alpha_{\beta_0}, \beta_0)) = w_0$. Then we note that for $\beta = 0$, $\mathcal{A}'_2(\alpha_\beta, \beta)$ coincides with $\mathcal{A}_2(\pi - 2)$, and in particular the induced worst-case cost becomes $3 + \pi$. Therefore $\text{Wrs}(\mathcal{A}'_2(\alpha_\beta, \beta))$ is 1-1 and onto to $[w_0, 3 + \pi]$ as β ranges in $[0, \beta_0]$.

Third, we study \mathcal{A}_2 , for which we know that $\text{Wrs}(\mathcal{A}_2(\pi - 2)) = 3 + \pi$. Again, the worst-case cost is monotone in α and $\mathcal{A}_2(0)$ coincides with benchmark algorithm \mathcal{B}_2 , which is $\text{Wrs}(\mathcal{A}_2(0)) = w_2$. Hence, $\text{Wrs}(\mathcal{A}_2(\alpha))$ is 1-1 and onto to $[3 + \pi, w_2]$ as α ranges in $[0, \pi - 2]$.

Finally, we argue that

$$\begin{aligned} \text{Avg}(\mathcal{A}_1(\alpha)) &\leq u_1(\alpha), \forall \alpha \in [1, \bar{\alpha}] \\ \text{Avg}(\mathcal{A}'_2(\alpha_\beta, \beta)) &\leq u'_2(\beta), \forall \beta \in [0, \beta_0] \\ \text{Avg}(\mathcal{A}_2(\alpha)) &\leq u_2(\alpha), \forall \alpha \in [0, \pi - 2] \end{aligned}$$

For this, we numerically compute $\text{Avg}(\mathcal{A}_1(\alpha))$, $\text{Avg}(\mathcal{A}'_2(\alpha_\beta, \beta))$, $\text{Avg}(\mathcal{A}_2(\alpha))$ for various values of parameters α, β , and we heuristically choose u_1, u'_2, u_2 so as to upper bound the average-case performance of $\mathcal{A}_1, \mathcal{A}'_2, \mathcal{A}_2$, effectively verifying our claim numerically. For each evacuation algorithm, we use Corollary 1, which together with the analytic description of our evacuation algorithms (see Definitions 4, 6, and 5) allow us to compute their average-case performance using computer-assisted calculations. Our numerical calculations are depicted in Figure 7.

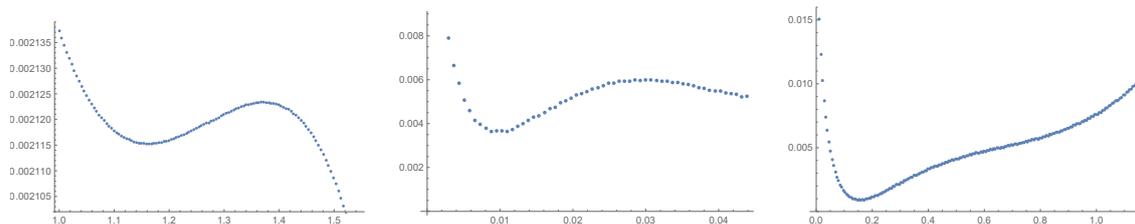


Figure 7. On the left $u_1(\alpha) - \text{Avg}(\mathcal{A}_1(\alpha))$, for $\alpha' \leq \alpha \leq \bar{\alpha}$. In the middle, $u'_2(\beta) - \text{Avg}(\mathcal{A}'_2(\alpha_\beta, \beta))$, for $0 \leq \beta \leq \beta_0$. On the right $u_2(\alpha) - \text{Avg}(\mathcal{A}_2(\alpha))$, for $0 \leq \alpha \leq \pi - 2$.

□

Finally, we aim to formally quantify the efficient frontier of our algorithms as depicted in Figure 1 (see Section 2.1). The parametric curves described in Theorem 6 provide, strictly speaking, an upper bound for the parametric curve of Figure 1. Next, we compute $g : \mathbb{R} \mapsto \mathbb{R}$, so that the parametric curves of Theorem 6 are written in the form $\{(g(w), w)\}_{w \in [w_1, w_2]}$. That would also imply that there is a solution to ${}_2\text{EVAC}_{F2F}^w$ of cost at most $g(w)$.

In that direction, we study each evacuation algorithm family $\mathcal{A}(p)$ with worst-case performance, say, $v(p)$, and average-case upper bound, say, $u(p)$. For each $w \in [w_1, w_2]$ in the range of $\mathcal{A}(p)$, we set $p = v^{-1}(w)$ so that the average-case performance achieved becomes $u(v^{-1}(w))$.

Recall that $\text{Wrs}(\mathcal{A}_i(\alpha)) = v(\alpha)$, so that $v^{-1}(w) = 1 + 2\pi - w$, and hence for algorithms \mathcal{A}_i we can easily compute $u_i(v^{-1}(w))$, $i = 1, 2$. For \mathcal{A}'_2 we recall that $\text{Avg}(\mathcal{A}'_2(\alpha_\beta, \beta))$ is decreasing in β . Since v_2^{-1} does not admit a closed form, we need to observe that $2.999 + \pi - \beta/2 \leq v_2(\beta) \leq 3 + \pi - \beta/2$ for all $\beta \in [0, \beta_0]$ so that an upper bound for $\text{Avg}(\mathcal{A}'_2(\alpha_\beta, \beta))$ admitting worst-case performance w can be computed by $u'_2(12.2812 - 2w)$.

Now for each $w \in [w_1, w_2]$ we need to specify which of the evacuation algorithms we will invoke. Please note that in Theorem 6 we chose the range of α in \mathcal{A}_1 to start from 1 so that as to guarantee that $\text{Wrs}(\mathcal{A}_1(1)) \geq w_0$. We note that $u'_2(12.2812 - 2w) = u_1(1 + 2\pi - w)$ for $w' \approx 6.12851$, so algorithm \mathcal{A}_1 should be invoked for $w \in [w_1, w']$ (and w' is obtained for $\alpha' := 1 + 2\pi - w' \approx 1.15468$), then \mathcal{A}'_2

for $w \in [w', 3 + \pi]$ (and w' is obtained for β' so that $v_2(\beta') = w'$, where $\beta' \approx 0.0241653$), and \mathcal{A}_2 for $w \in [3 + \pi, w_2]$. We conclude with the next Theorem (for convenience, the values of all constants are summarized at the end of Section 2.3).

Theorem 7. For every $w \in [w_1, w_2]$, the optimal solution to ${}_2\text{EVAC}_{F2F}^w$ is at most $g(w)$, where

$$g(w) = \begin{cases} -0.00889w^3 + 0.0248026w^2 + 0.338241w + 3.88629 & , w \in [w_1, w'] \ (\mathcal{A}_1(\alpha), \alpha \in [\alpha', \bar{\alpha}]) \\ -4245.38w^3 + 77893.3w^2 - 476397.w + 971235 & , w \in [w', 3 + \pi] \ (\mathcal{A}_2(\alpha_\beta, \beta), \beta \in [0, \beta']) \\ 0.093056w^2 - 1.70215w + 11.6328 & , w \in [3 + \pi, w_2] \ (\mathcal{A}_2(\alpha), \alpha \in [0, \pi - 2]) \end{cases}$$

7. Conclusions and Open Problems

We offered a new perspective to the well-studied problem of evacuating two robots from the disk, in the face-to-face model, first introduced in [1]. A series of results pertaining especially to the same domain have focused exclusively on deterministic algorithms and their worst-case (competitive) analyses. Our work can be understood as a first attempt to study the same problem in the realm of randomized algorithms. More specifically, in light of known positive results for the problem, we asked the question of minimizing the average-case performance of an evacuation algorithm, condition on that the worst-case performance remains bounded. We allowed that latter bound to range between the best known guarantee of a simple yet efficient evacuation algorithm (that was introduced in [1]) and that of a naive algorithm that simulates the optimal solution of one searcher. Our main contribution is the introduction of three new evacuation algorithms that perform well for different values of the worst-case bound, inducing a continuous bound of the efficient (pareto) frontier for the underlying multi-objective optimization problem (that of minimizing both the worst and the average-case performance). We also motivated our results by verifying, somehow surprisingly, that among the family of algorithms introduced in [27], which gave the second subsequent improvement of the worst-case performance, none of them could improve the average-case performance of the simple algorithm presented in [1]. In that sense our new algorithms outperform the best algorithms known for the problem, but in the new multi-objective setting. Our findings give rise to several open questions and directions that aim to better understand ${}_2\text{EVAC}_{F2F}^w$ as well as multi-objective optimization search-type problems.

Open questions specific to our problem ${}_2\text{EVAC}_{F2F}^w$ are:

- Prove lower bounds for ${}_2\text{EVAC}_{F2F}^w$, for any w . Is any of our algorithms, for any w optimal?
- For the value $w = \text{Wrs}(\mathcal{B}_1)$, we designed algorithm $\mathcal{A}_1(\alpha)$ for ${}_2\text{EVAC}_{F2F}^w$ which for a proper value of a has worst-case performance exactly w , while its average-case performance is strictly less than $\text{Avg}(\mathcal{B}_1)$. Is it feasible to attain worst-case performance strictly less than w , while having average-case performance at most $\text{Avg}(\mathcal{B}_1)$?
- The bound to the efficient (pareto) frontier we derived for problem ${}_2\text{EVAC}_{F2F}^w$ is indeed continuous, with respect to parameter w , but not differentiable. Is the optimal pareto frontier smooth, or is there any other family of algorithms that improves upon our results and gives a smooth transition between families of evacuation algorithms?
- The algorithmic families we derived for ${}_2\text{EVAC}_{F2F}^w$ exhibit the following property. \mathcal{A}_2 is a natural extension to \mathcal{B}_2 . Similarly, \mathcal{A}'_2 is a natural extension to \mathcal{A}_2 . Finally, \mathcal{A}_1 is a natural extension to \mathcal{B}_1 . However, \mathcal{A}_1 and \mathcal{A}'_2 have different behavior (there are no values of their parameters that induce the same evacuation protocol), even though for a proper choice of their parameters, they induce algorithms with the same worst-case and average-case performance.
- Observe that the average-case performance of \mathcal{B}_2 is $1 + \pi$. All our evacuation algorithms induce average cost at least $1 + \pi$. We conjecture that even in the wireless model, as well as for any number of robots, $1 + \pi$ is tight lower bound for the average performance of evacuation algorithm.

We conclude with some future directions that are inspired by our work and pertain to more general problems:

- Our algorithms can also be interpreted as randomized algorithms that have access to infinitely many bits (or enough many bits, in order to simulate a uniformly random deployment point on the circle). What if the algorithm has access only to a limited number of random bits?
- To the best of our knowledge, the current paper is the first attempt to study multi-objective optimization search-type problems. It was followed by [48,49] who considered time energy trade-offs for a search problems on the line. This line of research admits many future directions based on any combination of multiple objectives, e.g., worst-case, average-case and competitive cost, time, energy and any other efficiency measure, or even trade-offs involving number of faults or even complexity resources, e.g., memory, communication or randomness.

Author Contributions: Investigation, H.C. and P.S.; Supervision, K.G. All authors have read and agreed to the published version of the manuscript.

Funding: The second author and this research was supported in part by NSERC Discovery Grant.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Czyzowicz, J.; Gasieniec, L.; Gorry, T.; Kranakis, E.; Martin, R.; Pajak, D. Evacuating Robots via Unknown Exit in a Disk. In Proceedings of the DISC, Austin, TX, USA, 12–15 October 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 122–136.
2. Beck, A. On the linear search problem. *Isr. J. Math.* **1964**, *2*, 221–228. [[CrossRef](#)]
3. Bellman, R. An optimal search. *SIAM Rev.* **1963**, *5*, 274–274. [[CrossRef](#)]
4. Dobbie, J. A survey of search theory. *Oper. Res.* **1968**, *16*, 525–537. [[CrossRef](#)]
5. Benkoski, S.; Monticino, M.; Weisinger, J. A survey of the search theory literature. *Nav. Res. Logist. (NRL)* **1991**, *38*, 469–494. [[CrossRef](#)]
6. Stone, L. *Theory of Optimal Search*; Academic Press: New York, NY, USA, 1975.
7. Ahlswede, R.; Wegener, I. *Search Problems*; Wiley-Interscience: New York, NY, USA, 1987.
8. Alpern, S.; Gal, S. *The Theory of Search Games and Rendezvous*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2002; Volume 55.
9. Alpern, S.; Fokkink, R.; Gasieniec, L.; Lindelauf, R.; Subrahmanian, V. (Eds.) Ten Open Problems in Rendezvous Search. In *Search Theory: A Game Theoretic Perspective*; Springer: New York, NY, USA, 2013; pp. 223–230.
10. Albers, S.; Henzinger, M.R. Exploring unknown environments. *SIAM J. Comput.* **2000**, *29*, 1164–1188. [[CrossRef](#)]
11. Albers, S.; Kursawe, K.; Schuierer, S. Exploring unknown environments with obstacles. *Algorithmica* **2002**, *32*, 123–143. [[CrossRef](#)]
12. Deng, X.; Kameda, T.; Papadimitriou, C. How to learn an unknown environment. In Proceedings of the 32nd Annual Symposium of Foundations of Computer Science, San Juan, Puerto Rico, 1–4 October 1991; pp. 298–303.
13. Hoffmann, F.; Icking, C.; Klein, R.; Kriegel, K. The polygon exploration problem. *SIAM J. Comput.* **2001**, *31*, 577–600. [[CrossRef](#)]
14. Burgard, W.; Moors, M.; Stachniss, C.; Schneider, F.E. Coordinated multi-robot exploration. *IEEE Trans. Robot.* **2005**, *21*, 376–386. [[CrossRef](#)]
15. Thrun, S. A probabilistic on-line mapping algorithm for teams of mobile robots. *Int. J. Robot. Res.* **2001**, *20*, 335–363. [[CrossRef](#)]
16. Yamauchi, B. Frontier-based exploration using multiple robots. In Proceedings of the Second International Conference on Autonomous Agents, Minneapolis, MI, USA, 9–13 May 1998; ACM: New York, NY, USA, 1998; pp. 47–53.
17. Kleinberg, J. On-line search in a simple polygon. In Proceedings of the SODA, Arlington, VA, USA, 23–25 January 1994; p. 8.

18. Mitchell, J.S. Geometric shortest paths and network optimization. *Handb. Comput. Geom.* **2000**, *334*, 633–702.
19. Papadimitriou, C.H.; Yannakakis, M. Shortest paths without a map. In *ICALP*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 610–620.
20. Chung, T.H.; Hollinger, G.A.; Isler, V. Search and pursuit-evasion in mobile robotics. *Auton. Robot.* **2011**, *31*, 299–316.
21. Fomin, F.V.; Thilikos, D.M. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.* **2008**, *399*, 236–245.
22. Lidbetter, T. Hide-and-Seek and other Search Games. Ph.D. Thesis, The London School of Economics and Political Science (LSE), London, UK, 2013.
23. Nahin, P. *Chases and Escapes: The Mathematics of Pursuit and Evasion*; Princeton University Press: Princeton, NJ, USA, 2012.
24. Baumann, N.; Skutella, M. Earliest arrival flows with multiple sources. *Math. Oper. Res.* **2009**, *34*, 499–512.
25. Fekete, S.; Gray, C.; Kröller, A. Evacuation of rectilinear polygons. In *Combinatorial Optimization and Applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 21–30.
26. Czyzowicz, J.; Georgiou, K.; Kranakis, E.; Narayanan, L.; Opatrny, J.; Vogtenhuber, B. Evacuating Robots from a Disc Using Face to Face Communication. In Proceedings of the CIAC 2015, Paris, France, 20–22 May 2015; Springer: Berlin/Heidelberg, Germany, 2015.
27. Brandt, S.; Laufenberg, F.; Lv, Y.; Stolz, D.; Wattenhofer, R. Collaboration without Communication: Evacuating Two Robots from a Disk. In Proceedings of the Algorithms and Complexity—10th International Conference, CIAC, Athens, Greece, 24–26 May 2017; pp. 104–115.
28. Disser, Y.; Schmitt, S. Evacuating two robots from a disk: A second cut. In *International Colloquium on Structural Information and Communication Complexity*; Lecture Notes in Computer Science; Censor-Hillel, K., Flammini, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11639, pp. 200–214.
29. Czyzowicz, J.; Dobrev, S.; Georgiou, K.; Kranakis, E.; MacQuarrie, F. Evacuating two robots from multiple unknown exits in a circle. *Theor. Comput. Sci.* **2018**, *709*, 20–30.
30. Pattanayak, D.; Ramesh, H.; Mandal, P.S.; Schmid, S. Evacuating Two Robots from Two Unknown Exits on the Perimeter of a Disk with Wireless Communication. In Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN 2018, Varanasi, India, 4–7 January 2018; Bellavista, P., Garg, V.K., Eds.; ACM: New York, NY, USA, 2018; pp. 20:1–20:4.
31. Chuangpishit, H.; Mehrabi, S.; Narayanan, L.; Opatrny, J. Evacuating equilateral triangles and squares in the face-to-face model. *Comput. Geom.* **2020**, *89*, 101624.
32. Czyzowicz, J.; Kranakis, E.; Krizanc, D.; Narayanan, L.; Opatrny, J.; Shende, S. Wireless Autonomous Robot Evacuation from Equilateral Triangles and Squares. In Proceedings of Ad-hoc, Mobile, and Wireless Networks, ADHOC-NOW, Athens, Greece, 29 June–1 July 2015; pp. 181–194.
33. Brandt, S.; Foerster, K.T.; Richner, B.; Wattenhofer, R. Wireless evacuation on m rays with k searchers. *Theor. Comput. Sci.* **2020**, *811*, 56–69.
34. Angelopoulos, S.; Dürr, C.; Lidbetter, T. The expanding search ratio of a graph. *Discret. Appl. Math.* **2019**, *260*, 51–65.
35. Borowiecki, P.; Das, D.; Dereniowski, D.; Kuszner, L. Distributed Evacuation in Graphs with Multiple Exits. In *Structural Information and Communication Complexity, Proceedings of the 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, 19–21 July 2016*; Revised Selected Papers, Lecture Notes in Computer Science; Suomela, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9988, pp. 228–241.
36. Chrobak, M.; Gasieniec, L.T.G.; Martin, R. Group Search on the Line. In *SOFSEM 2015*; Springer: Berlin/Heidelberg, Germany, 2015.
37. Georgiou, K.; Lucier, J. Weighted Group Search on a Line. In Proceedings of the 16th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS, Pisa, Italy, 7–11 September 2020.
38. Baeza Yates, R.; Culberson, J.; Rawlins, G. Searching in the plane. *Inf. Comput.* **1993**, *106*, 234–252.
39. Czyzowicz, J.; Georgiou, K.; Godon, M.; Kranakis, E.; Krizanc, D.; Rytter, W.; Włodarczyk, M. Evacuation from a disc in the presence of a faulty robot. In *International Colloquium on Structural Information and Communication Complexity*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 158–173.

40. Georgiou, K.; Kranakis, E.; Leonardos, N.; Pagourtzis, A.; Papaioannou, I. Optimal Cycle Search Despite the Presence of Faulty Robots. In *Algorithms for Sensor Systems, Proceedings of the 15th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2019, Munich, Germany, 12–13 September 2019*; Revised Selected Papers, Lecture Notes in Computer Science; Dressler, F., Scheideler, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11931, pp. 192–205.
41. Pattanayak, D.; Ramesh, H.; Mandal, P.S. Chauffeur-ing a Crashed Robot from a Disk. In *Algorithms for Sensor Systems, Proceedings of the 15th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2019, Munich, Germany, 12–13 September 2019*; Revised Selected Papers, Lecture Notes in Computer Science; Dressler, F., Scheideler, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11931, pp. 177–191.
42. Bonato, A.; Georgiou, K.; MacRury, C.; Pralat, P. Probabilistically Faulty Searching on a Half-Line. In *Proceedings of the 14th Latin American Theoretical Informatics Symposium, University of Sao Paulo, Sao Paulo, Brazil, 25–29 May 2020*.
43. Georgiou, K.; Kranakis, E.; Steau, A. Searching with Advice: Robot Fence-Jumping. *J. Inf. Process.* **2017**, *25*, 559–571.
44. Czyzowicz, J.; Georgiou, K.; Killick, R.; Kranakis, E.; Krizanc, D.; Narayanan, L.; Opatrny, J.; Shende, S. Priority Evacuation from a Disk: The case of $n \geq 4$. *Theor. Comput. Sci.* **2020**, accepted.
45. Czyzowicz, J.; Georgiou, K.; Killick, R.; Kranakis, E.; Krizanc, D.; Narayanan, L.; Opatrny, J.; Shende, S.M. Priority evacuation from a disk: The case of $n = 1, 2, 3$. *Theor. Comput. Sci.* **2020**, *806*, 595–616. [[CrossRef](#)]
46. Georgiou, K.; Karakostas, G.; Kranakis, E. Search-and-Fetch with One Robot on a Disk—(Track: Wireless and Geometry). In *Algorithms for Sensor Systems, Proceedings of the 12th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2016, Aarhus, Denmark, 25–26 August 2016*; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2016; pp. 80–94.
47. Georgiou, K.; Karakostas, G.; Kranakis, E. Search-and-Fetch with 2 Robots on a Disk: Wireless and Face-to-Face Communication Models. *Discret. Math. Theor. Comput. Sci.* **2019**, *21*. [[CrossRef](#)]
48. Czyzowicz, J.; Georgiou, K.; Killick, R.; Kranakis, E.; Krizanc, D.; Lafond, M.; Narayanan, L.; Opatrny, J.; Shende, S. Energy Consumption of Group Search on a Line. In *Leibniz International Proceedings in Informatics (LIPIcs), Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019), Patras, Greece, 8–12 July 2019*; Baier, C., Chatzigiannakis, I., Flocchini, P., Leonardi, S., Eds.; Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2019; Volume 132, pp. 137:1–137:15. [[CrossRef](#)]
49. Kranakis, E.J.; Krizanc, D.K.; Georgiou, M.L.; Killick, R.; Narayanan, L.; Opatrny, J.; Shende, S. Time-Energy Tradeoffs for Evacuation by Two Robots in the Wireless Model. In *Structural Information and Communication Complexity, Proceedings of the 26th International Colloquium, SIROCCO 2019, L’Aquila, Italy, 1–4 July 2019*; Lecture Notes in Computer Science; Censor-Hillel, K., Flammini, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11639, pp. 185–199.
50. Lamprou, I.; Martin, R.; Schewe, S. Fast two-robot disk evacuation with wireless communication. In *Proceedings of the DISC, Paris, France, 27–29 September 2016*; pp. 1–15.
51. Czyzowicz, J.; Kranakis, E.; Krizanc, D.; Narayanan, L.; Opatrny, J.; Shende, S.M. Linear Search with Terrain-Dependent Speeds. In *Algorithms and Complexity, Proceedings of the 10th International Conference, CIAC 2017, Athens, Greece, 24–26 May 2017*; Lecture Notes in Computer Science; Fotakis, D., Pagourtzis, A., Paschos, V.T., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10236, pp. 430–441.
52. Czyzowicz, J.; Georgiou, K.; Kranakis, E. Group Search and Evacuation. In *Distributed Computing by Mobile Entities; Current Research in Moving and Computing*; Flocchini, P., Prencipe, G., Santoro, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Chapter 14, pp. 335–370.
53. López-Ortiz, A.; Sweet, G. Parallel searching on a lattice. In *Proceedings of the CCCG, 13–15 August 2001*; pp. 125–128.
54. Emek, Y.; Langner, T.; Uitto, J.; Wattenhofer, R. Solving the ANTS Problem with Asynchronous Finite State Machines. In *Automata, Languages, and Programming, Proceedings of the 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, 8–11 July 2014*; Lecture Notes in Computer Science, Part II; Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8573, pp. 471–482.

55. Lenzen, C.; Lynch, N.; Newport, C.; Radeva, T. Trade-offs Between Selection Complexity and Performance when Searching the Plane Without Communication. In *ACM Symposium on Principles of Distributed Computing, Proceedings of the PODC '14, Paris, France, 15–18 July 2014*; HalldÃ³rsson, M.M., Dolev, S., Eds.; ACM: New York, NY, USA, 2014; pp. 252–261.
56. Acharjee, S.; Georgiou, K.; Kundu, S.; Srinivasan, A. Lower Bounds for Shoreline Searching With 2 or More Robots. In *Proceedings of the 23rd International Conference on Principles of Distributed Systems (OPODIS'19), Neuchâtel, Switzerland, 17–19 December 2019*; Felber, P., Friedman, R., Gilbert, S., Miller, A., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Dagstuhl, Germany, 2019; Volume 153, pp. 26:1–26:11.
57. Dobrev, S.; Kralovic, R.; Pardubska, D. Improved Lower Bounds for Shoreline Search. In *Structural Information and Communication Complexity, Proceedings of the 27th International Colloquium, SIROCCO 2020, Paderborn, Germany, 29 June–1 July 2020*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020.
58. Miettinen, K. Nonlinear multiobjective optimization. In *International Series in Operations Research and Management Science*; Kluwer: Dordrecht, The Netherlands, 1998; Volume 12, pp. 1–298.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).