Maximum Thick Paths in Static and Dynamic Environments*

Esther M. Arkin† Joseph S. B. Mitchell* Valentin Polishchuk‡

Abstract

We consider the problem of finding a large number of disjoint paths for unit disks moving amidst static or dynamic obstacles. The problem is motivated by the capacity estimation problem in air traffic management, in which one must determine how many aircraft can safely move through a domain while avoiding each other and avoiding "no-fly zones" and predicted weather hazards. For the static case we give efficient exact algorithms, based on adapting the "continuous uppermost path" paradigm. As a by-product, we establish a continuous analogue of Menger's Theorem.

Next we study the dynamic problem in which the obstacles may move, appear and disappear, and otherwise change with time in a known manner; in addition, the disks are required to enter/exit the domain during prescribed time intervals. Deciding the existence of just one path, even for a 0-radius disk, moving with bounded speed is NP-hard, as shown by Canny and Reif [1]. Moreover, we observe that determining the existence of a given number of paths is hard even if the obstacles are static, and only the entry/exit time intervals are specified for the disks. This motivates studying "dual" approximations, compromising on the radius of the disks and on the maximum speed of motion.

Our main result is a pseudopolynomial-time dual-approximation algorithm. If K unit disks, each moving with speed at most 1, can be routed through an environment, our algorithm finds (at least) K paths for disks of radius somewhat smaller than 1 moving with speed somewhat larger than 1.

1 Introduction

Path planning in geometric domains is an important computational geometry subject with applications in robotics, VLSI routing, air traffic management (ATM), sensor networks, etc., [2]. In many applications it is of interest to find *multiple* disjoint paths for *non-point* objects avoiding *moving* obstacles. This is the problem studied in this paper.

The input to the problem is specified by a polygonal domain, with two edges of the outer polygon designated as the "source" and the "sink". The holes/obstacles in the domain move along known trajectories. Also given is an interval (or, more generally, a finite set of intervals), T_I specifying the allowed times for disks to enter the domain; any disk may enter during any time in T_I. Similarly, T_O specifies the allowed times for disks to exit the domain. The goal is to find a maximum number of trajectories for congruent disks, moving with bounded speed, entering (resp., exiting) the domain through the source (resp., sink) during T_I (resp., T_O), never intersecting each other, nor the obstacles. The diameter of the disks determines the minimum separation between points during their motions along the trajectories.

Motivation

We are motivated by an application in ATM; similar problems may arise in other coordinated motion planning problems in transportation engineering, e.g., shipping vessels, robotic material handling machines, etc.

*A preliminary version of this paper appears in the *Proc. 24th Annual ACM Symposium on Computational Geometry*, pp. 20–27, 2008.

†Stony Brook University, {estie,jsbm}@ams.stonybrook.edu

‡Corresponding author. Helsinki Institute for Information Technology, University of Helsinki and Helsinki University of Technology, valentin.polishchuk@cs.helsinki.fi

The polygon P models an airspace through which aircraft intend to fly. In ATM applications, P may specify the boundary of a "sector" or a"center" (set of sectors) in the National Airspace System, or a region of interest, "flow constrained area", for traffic flow management. We assume that the aircraft remain at constant altitude (as is often the case during en route flight), so that we can consider the problem to be in a two-dimensional domain. There is often a dominant flow direction at a given altitude (since, e.g., west-to-east air traffic is altitude-separated from east-to-west traffic); thus, we consider all aircraft to be entering through one side of P and exitting through another side of P.

When hazardous weather conditions are present, portions of the airspace are effectively blocked (at the given altitude) by regions of intense weather activity that serve as (moving) obstacles (generally called "constraint" in the ATM community). There are static obstacles within P that correspond to "no-fly" zones arising from special use airspace (military airspace, noise abatement zones, security zones over cities, etc). Weather forecasts over time specify moving obstacles; often there is an additional safety buffer placed around weather hazards to account for a safe separation and the fact that the forecasts are inaccurate.

We are interested in determining the "capacity" of the airspace P: How many aircraft can safely be routed through P, during a specified time window, while maintaining safe separation from each other and from the obstacles? Of course, an air traffic controller will never risk sending the maximum number of aircraft possible; rather, knowledge of the maximum number is relevant to estimating the safe capacity of the airspace (likely a small fraction of the maximum). Further, a set of feasible routes that achieve maximum, or close to maximum, capacity may suggest a set of possibilities that a controller can use to route traffic around hazards.

Related Work

Time-dependent flows in *graphs* have been addressed with "time expansion" [3], a technique similar to ours (slicing the time and building a "layered" graph with one layer per slice).

In the static setting, finding a maximum number of source-sink paths in a graph is equivalent to computing a maximum flow. Three classical graph-theoretic results are the MaxFlow-MinCut, the Flow Decomposition, and Menger's Theorem (see, e.g., [4]). The extensions of these theorems to (static) geometric domains are developed in [5, 6, 7] and in this paper. Applications of some of these techniques to weather hazard avoidance in ATM are reported in [8, 9, 10].

A lot of research has been done on computing *fastest* paths amidst moving/morphing obstacles for a point object, possibly with nonholonomic motion constraints, both in 2D and 3D [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. See the books [24, 25], Section 4.4 in the survey [26], and references therein for details. In this paper we are concerned only with the *existence* of paths; other than respecting the entry/exit time intervals, the disks are free to move slowly (although, intuitively, in an optimal solution, disks will tend to stay in the domain for a short period of time in order that they not occupy the time/space needed for the other disks). Existence of a single path avoiding a set of obstacles was addressed in [27].

Approaches for planning paths for multiple objects fall roughly into two main categories: *prioritized*, when the paths are routed one-by-one, and *coordinated*, ranging from centralized to roadmap-based to decoupled (the classification is taken from [28], where a more detailed discussion of the approaches is given). Existing algorithms are *heuristic*; in ATM applications, one method is implemented in the Flow-Based Route Planner (FBRP) [29]. The FBRP searches for the paths greedily and iteratively: each computed "thick" path becomes a constraint (obstacle) in space-time for subsequent paths. Although examples exist for which arbitrarily many paths may be routed, while the FBRP produces only one, the planner performs very well in practical situations [8, 9, 30, 29, 31].

Our Contribution

Similar to existing heuristics, we employ a discretization of time and space. We consider a hexagonal grid (packing) of disks at each time slice, and remove each disk that intersects the set of obstacles. We then compute a maximum flow in a graph built on the disks; the flow decomposes into a set of paths. We prove that the conflicts between the paths, introduced by the discretization, can be resolved locally. We show how

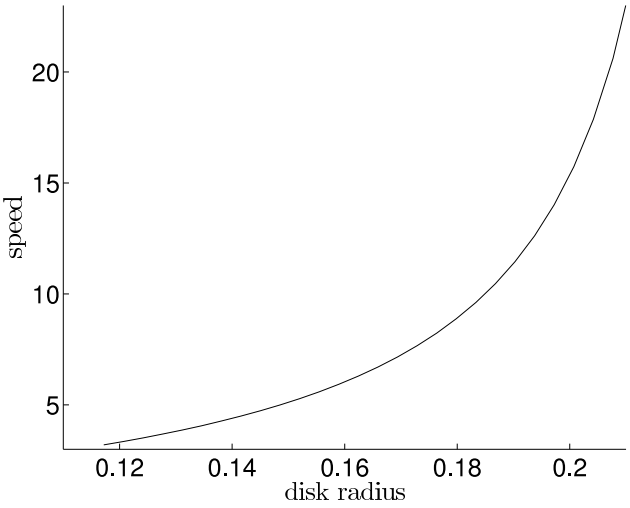

Figure 1: The tradeoff between the radius of the disks and the speed limit.

to balance the time discretization step, the radius of the disks packed, and the disks' speed: we prove that if there exist K paths through the domain for unit disks moving with speed at most 1, our algorithm will find, for any $\Delta t \leq 1/3$, (at least) K paths for disks of radius $\sqrt{3} - 1.5 + (2.25 - 1.5\sqrt{3})\Delta t \approx .23 - .35\Delta t$, moving with speed at most $1.1/\Delta t - 0.8$ (Fig. 1). For instance, for $\Delta t = 1/4$ the radius of the disks is ~ 0.14 and the speed limit is 3.6. We give hardness results that justify the need for approximate solutions.

Our algorithm runs in pseudopolynomial time. (Note that the *number* of paths that exist in a domain may be exponential in the input size, e.g., there may exist $\Omega(N)$ paths in a $2 \times N$ rectangle — specified with $O(\log N)$ bits.) Despite the fact that the analysis of the algorithm's correctness is somewhat involved, the algorithm itself can be easily implemented using only basic tools from computational geometry.

For the case of static obstacles, we give an exact polynomial-time algorithm to find a maximum number of "thick" non-crossing paths in a polygonal domain. Our solution is based on the continuous-Dijkstra-type uppermost path algorithm for finding a maximum flow in geometric environments [6]. As a by-product, we formulate and prove the Continuous Menger's Theorem, an extension of the classic graph theorem to geometric domains.

2 Static Obstacles

We begin with a formal statement of our problem and a review of some relevant notions and results from previous work [6, 7]. The input to the problem is a polygonal domain Ω specified by an outer (simple) polygon P and a set $\mathcal{H}$ of holes in it. Let n denote the number of vertices on the boundary of Ω, and let h denote the number of holes. Two edges, Γ_I and Γ_O, of P are designated as the *source* and the *sink*, respectively. As is standard in the thick paths and continuous flow literature [6, 7, 32], we assume that P is augmented with Riemann sheets attached along Γ_I and Γ_O. The purpose of the sheets is to allow paths (disks) to enter the domain from the sheets, without intersecting the boundary of P, even if when projected onto the base sheet, $\mathbb{R}^2$, the disks would intersect the boundary (Fig. 2).

Thick Paths. A (thin) path is a simple (non-self-intersecting) source-sink curve in Ω. For $w \geq 0$ and $S \subset \mathbb{R}^2$ let $\langle S \rangle^{(w)}$ denote the Minkowski sum of S and the disk of radius w centered at the origin. A *w-thick path* $\Pi \subset \Omega$ is the Minkowski sum of a path π and the disk of radius w: $\Pi = \langle \pi \rangle^{(w)}$. The path π is called the *reference path* for the thick path Π. In a simple polygon, a w-thick path can be found in linear time. A 1-thick path is called just a *thick* path. For $w \in \mathbb{N}$, a w-thick path is a representation of a set of w thick paths. (This statement is actually about the "canonical parts" of the thick paths.) See Fig. 3 for an illustration

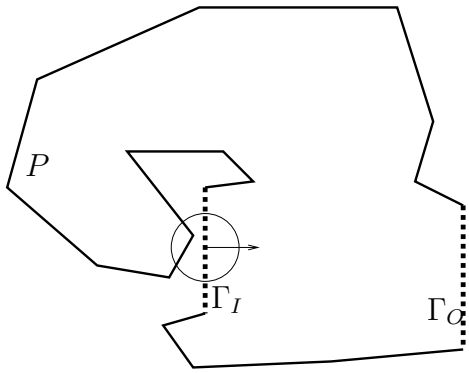

Figure 2: A disk legally enters the domain from the Riemann sheet.

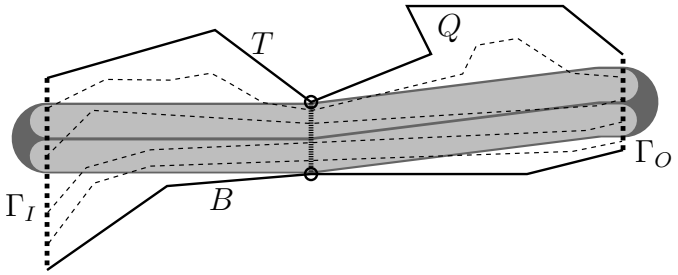

Figure 3: Source and sink edges Γ_I and Γ_O (thick, dashed) split the boundary of a simple polygon Q into top T and bottom B. A maximum flow has value 4 (thin, dashed streamlines), which is equal to the length of a mincut (the segment connecting the two hollow circles). The support of a minimum-cost flow of value 4 is a 2-thick path (grey), which may be viewed as the union of two 1-thick paths (light grey).

and [7] for details.

Minimum Cut. The source and sink edges split the boundary of P into two parts, the *top* T and the *bottom* B (see Fig. 3). A *mincut* through Ω is a shortest path in Ω from T to B in the *0/1 metric* [33]: The cost per unit length of travel is 0 within the holes and 1 within Ω. See [6] for details.

Flows in the Continuum. A *flow* in Ω is a vector field $\sigma : \Omega \mapsto \mathbb{R}^2$ such that: (1) $\forall x \in \Omega$, $\operatorname{div} \sigma(x) = 0$, i.e., there are no sources/sinks inside Ω; (2) $\forall x \in \partial\Omega \setminus \Gamma_I \setminus \Gamma_O$, $\sigma(x) \cdot \mathbf{n}(x) = 0$, where $\mathbf{n}(x)$ is the outward pointing unit vector normal to $\partial\Omega$ at x, i.e., the flow penetrates $\partial\Omega$ only at $\Gamma_I \cup \Gamma_O$; (3) $\forall x \in \Omega$, $|\sigma(x)| \leq 1$, i.e., each point in Ω has *unit capacity*. The *value* of the flow σ is defined as $V = \int_{\Gamma_O} \sigma \cdot \mathbf{n}\, ds$ $(= -\int_{\Gamma_I} \sigma \cdot \mathbf{n}\, ds)$. The *cost* of the flow is the area of the flow's support.

MinCut and Thick Paths in a Simple Polygon. The MaxFlow-MinCut and Flow Decomposition Theorems are classical network flow results [4]. Their geometric counterparts were developed in [5, 6] and [7]. The Continuous MaxFlow-MinCut Theorem states that the length of a mincut is equal to the maximum value of a flow in Ω. The Continuous Flow Decomposition Theorem asserts that the support of a minimum-cost flow can be decomposed into a set of thick paths.

For a simple polygon Q, the length of a mincut is just the distance between T and B. From the results in [6, 7], reviewed above, we have:

Lemma 2.1. *Let d be the distance between T and B. There exists a $\lfloor d/2 \rfloor$-thick path in Q. The path can be found in linear time. The path is a representation of a set of $\lfloor d/2 \rfloor$ thick paths.*

A Thick Path May Self-Overlap. By definition, the reference path of a thick path is simple, i.e., non-self-intersecting. At the same time, the definition allows a thick path to be self-overlapping. In fact, in some

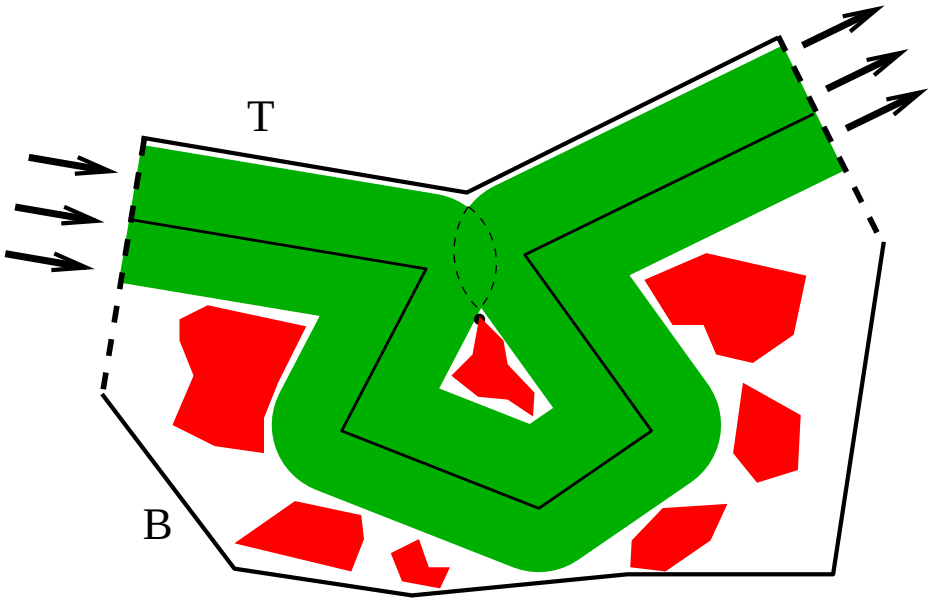

Figure 4: Only a self-overlapping thick path exists.

instances only self-overlapping thick paths exist (see Fig. 4 and Fig. 4 in [34]). This means, in particular, that routing a thick path is different from routing a thick *wire*—a non-self-overlapping thick path. While a thick path can be found by the standard procedure of offsetting the obstacles and then searching for a thin path within the modified free space, we are not aware of *any* algorithm for finding a thick wire. To the best of our knowledge, this difference between paths and wires has not been noticed before except in [34].

Continuous Uppermost Paths. The maximum flow algorithm of [6] uses the following "grass fire" analogy. Suppose that the free space is grass over which fire travels at speed 1. Suppose also that the holes are composed of a highly flammable material (the fire travels through a hole at infinite speed) so that as soon as the fire hits a hole, its entire boundary immediately ignites. Ignite T at time 0. The *wavefront* at time τ is the boundary of the grass that has burned by τ. The wavefronts make up the streamlines of the flow. The algorithm fills up the free space with the streamlines as the fire propagates until reaching B, at which time no more streamlines can be found. The *events* of the algorithm happen when the fire hits an obstacle. Two things happens at an event: the obstacle's boundary joins the wavefront, and the streamlines start going below the obstacle. Refer to Fig. 5.

2.1 The Algorithm

Our method to compute the maximum number of thick paths in Ω uses a modification of the continuous uppermost path algorithm. The modification involves attaching, where necessary, Riemann flaps to allow for a self-overlapping thick path to remain non-self-overlapping in the modified domain.

Specifically, suppose that the fire has not hit a hole after burning for 2 units of time. Then we route a thick path inside the burned grass. Lemma 2.1 ensures that a path exists. We argue below that routing a path as described cannot hurt the existence of the other paths in a collection of the maximum number of thick paths. We then start over, treating the wavefront at 2 as the new T. Refer to Fig. 6.

Suppose, on the other hand, that a hole H is hit by the fire at time $\tau < 2$. Let e be the segment of length τ that connects H to T; we assume that e is unique. We slit Ω along e and around ∂H. This "carves out" H, and it is no longer a hole (since it is now *outside* the domain). We attach a Riemann sheet to each copy of e. In each sheet, we place a circular segment, of radius 2, having e as a chord. Refer to Fig. 7. Now we continue the grass fire, not only igniting the whole boundary of H, but also a belt of thickness τ around it.

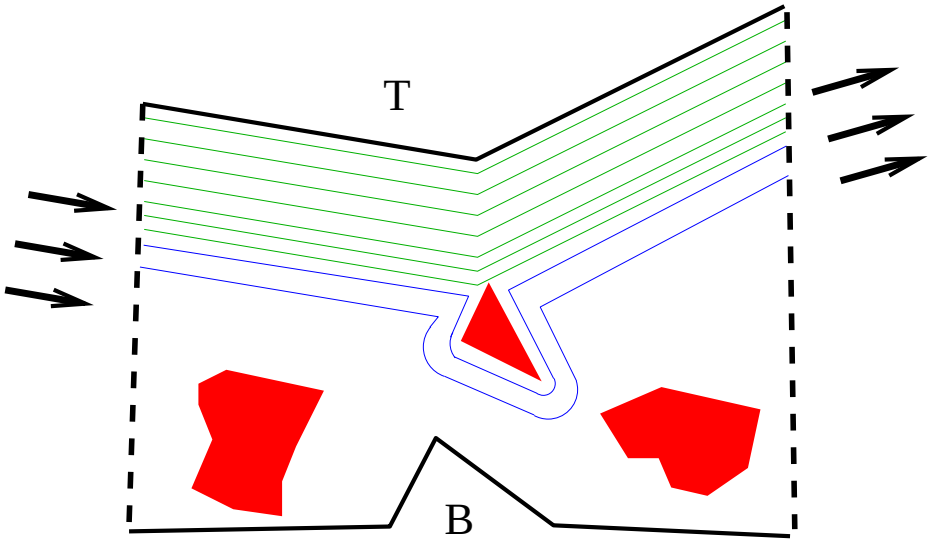

Figure 5: The wavefronts make up the streamlines of the flow. After the wavefront hits an obstacle, the streamlines start going around it.

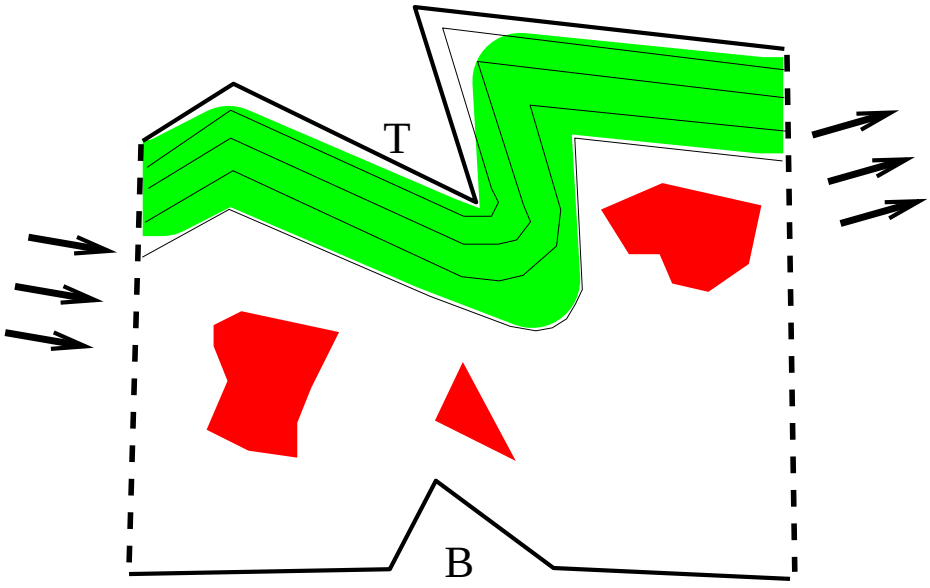

Figure 6: The event time is greater than 2: By Lemma 2.1 a thick path can be routed within the grass that is burned by time 2.

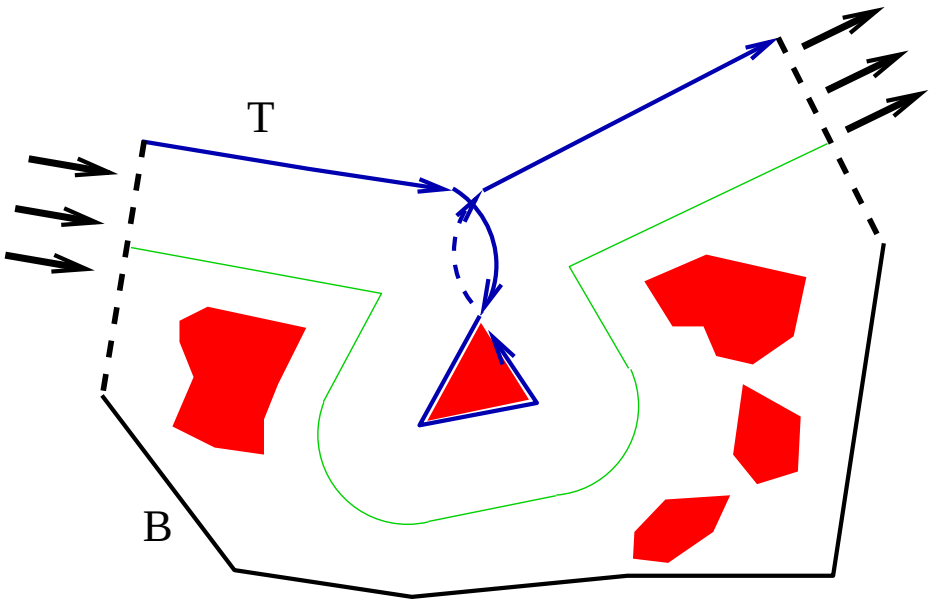

Figure 7: The event time is less than 2: Slit $\mathbb{R}^2$ along the shortest segment connecting the hit hole to T, attach Riemann flaps along each copy of the slit edge, and place a circular segment inside each flap. The new domain excludes the hole. The arrows indicate clockwise traversal of the new top. The new wavefront is shown (dashed-dotted) with its distance from the hole equal to the event time.

This belt represents the fact that the path that is being routed "jumps over" the hole and runs now on the other side of it. As before, when the grass has burned for 2 units of time without hitting a hole, we route a thick path within the burned grass. Carving out the holes and attaching the circular segments ensures that we are routing within a simple polygon; thus, Lemma 2.1 may be applied to show the existence of a thick path in it. We argue that, when projected to the base sheet $\mathbb{R}^2$ (where Ω resides), the reference path of the routed thick path does not intersect itself, and the thick path does not intersect any holes. As before, it can be argued that the routed path does not interfere with the other thick paths to be routed. Also as before, we treat the wavefront as the new top, and continue the propagation.

Remark 1. Observe that we cannot simply bridge H to the top. Indeed, while no thick path can have H below it (since the distance from H to T is less than 2), parts of a thick path may run "between" H and T. Moreover, different thick paths may occupy overlapping portions of the space between H and T (see Fig. 8). This shows that it is not possible to build a "bridge" between H and T such that *no* thick path would ever cross the bridge. Furthermore, as the example in Fig. 4 shows, just one thick path may completely "cut off" H from T so that, although H is above the path, there is no path (bridge) in Ω between the hole and the top.

As described above, the running time of our algorithm is proportional to the output size, i.e., to the maximum number, K, of thick paths in Ω. We can remove this dependency by stopping the fire propagation *only* when a hole is hit. We give the details in the next theorem.

Theorem 2.2. *(A representation of) the maximum number of disjoint thick paths can be found in* $O(nh + n \log n)$ *time.*

Proof. Let τ^* be the time of the first event, i.e., the time when the fire reaches a hole, H, in Ω. Suppose that $\tau^* \geq 2$; let $W = \lfloor \tau^*/2 \rfloor \geq 1$. Let π be the wavefront at $2W \leq \tau^*$. The part of Ω between T and π is a simple polygon, Ω_π, in which a flow of value $2W$ exists; this follows from the maxflow algorithm of [6] (streamlines correspond to wavefronts). By Lemma 2.1, there exists a W-thick path Π in Ω_π, and the path is a representation of a set of W thick paths. We continue the propagation, treating π as the new top.

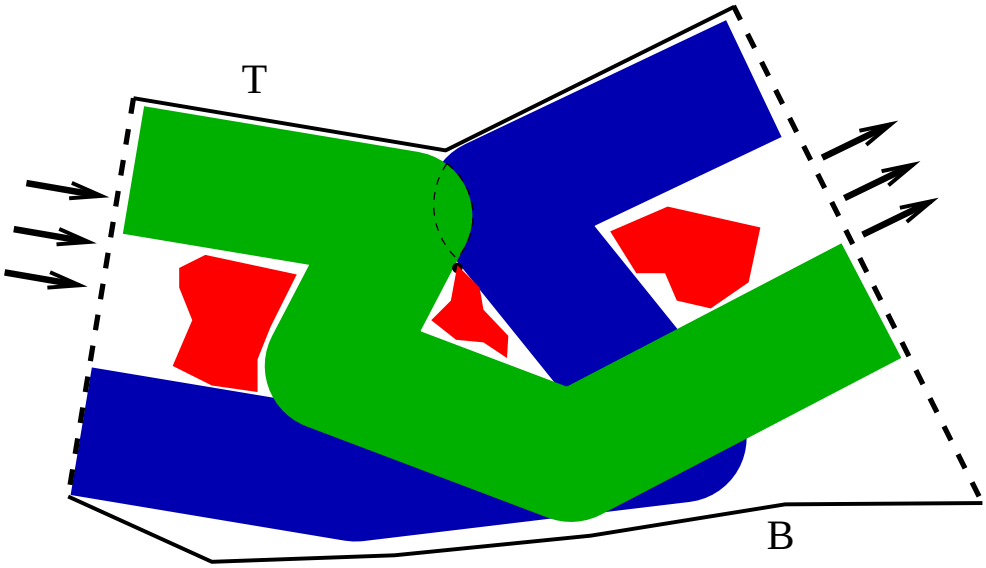

Figure 8: There is no unique bridge to T that would not be crossed by *any* thick path; different paths may "cut off" the hole from the top differently.

Suppose now that $\tau^* < 2$. Since the distance from H to T is less than 2, the reference path of a thick path cannot intersect e (the shortest segment between H and T). Thus, the reference path of a thick path, which will be routed after the fire burns eventless for more than 2 units of time, does not intersect itself even if parts of the thick path intersect e. The parts of a thick path that could possibly intersect e belong to the circular segments, of radius 2, that have e as a chord (Fig. 7). The parts cannot intersect an obstacle, since no obstacle intersects any of the circular segments. Indeed, the diameter of each of the segments is $\tau^* < 2$; thus, if a hole intersected a segment, the intersection point (and not the endpoint of e) would have been the event point. Overall, this shows that even if the thick path "bulges into" the circular segments, when projected onto the base sheet $\mathbb{R}^2$, it remains a valid thick path.

To prove correctness of the algorithm we need to show that the W (uppermost) thick paths produced by the algorithm after an event with $\tau^* \geq 2$ do not "block" routing (in the future) of the remaining $K - W$ thick paths. This follows from the following fact: Let $(\Pi_1, \ldots, \Pi_K)$ be a set of K thick non-crossing paths in Ω; then, the path Π_{W+1} does not intersect Ω_π. Indeed, the distance from π to T is $2W$; thus, if π were intersected by (the upper boundary of) Π_{W+1}, the W thick paths could not have "fit in between" Π_{W+1} and T.

Considering the running time, the grass fire can be simulated in $O(nh + n \log n)$ time, as described in [6]. There are $O(h)$ events. If $\tau^* < 2$, the modifications to the domain take constant time per event. Otherwise, W-thick paths can be routed (in a simple polygon) in linear time (Lemma 2.1).

The algorithm produces $K^* \leq h + 1$ thick paths. Indeed, every time a new path is routed, some hole that was previously below the routed paths becomes a hole above the subsequent paths. Using the thick paths output by the algorithm, one can find, for any $k \in \{1 \ldots K\}$, the kth path Π_k in the collection of K thick paths in time proportional to the combinatorial complexity of Π_k. In particular, all K paths may be output in $O(Kn)$ time. See [7] for the details. □

Remark 2. The running time of our algorithm matches that of the algorithm for the maxflow [6]. The bottleneck in both algorithms is the grass fire propagation in the "0/1 metric" [33]. It is possible that the $O(n \log n)$-time continuous Dijkstra algorithm of Hershberger and Suri [35] for shortest paths in polygonal domains can be extended to the 0/1 metric.

Thresholded Critical Graph and the Continuous Menger's Theorem. The notion of the *critical graph* of the domain [33, 6] is central to finding shortest paths in the 0/1 metric and to computing mincuts and maxflows in geometric domains. The critical graph has a vertex for each hole in $\mathcal{H}$; the length of an edge (i, j) is the distance between the holes H_i and H_j in the 0/1 metric (it is enough to connect by edges only those holes whose shortest connecting segment does not intersect other holes). Mitchell [6] showed that the length of a mincut, and hence the value of a maxflow, in the domain is equal to the length of a shortest T-B path in the critical graph.

Menger's Theorem states that for two vertices s, t of a graph, the maximum number of (internally) vertex-disjoint s-t paths in the graph is equal to the minimum number of vertices whose removal disconnects s and t. The results in this section establish the continuous version of the theorem. To state the theorem, we introduce the *thresholded critical graph* $G_{\lfloor\rfloor}$, which is the critical graph in which the length of each edge is divided by 2 and rounded down to the nearest integer.

Theorem 2.3. [The Continuous Menger's Theorem] *The maximum number of thick non-crossing paths in a domain is equal to the length of a shortest T-B path in the thresholded critical graph.*

Proof. The events in our uppermost path algorithm correspond to the wavefront reaching the holes. The number of paths routed at each event is equal to the length of an edge in the thresholded critical graph. By induction on the ordinal number of the event, the number of paths routed by the time of the event is equal to the shortest-path distance from T to the hit hole in $G_{\lfloor\rfloor}$. In particular, the total number of paths routed by the algorithm is equal to the shortest T-B path in the critical graph. □

3 Moving Obstacles

For the reminder of the paper we consider the case when the positions and shapes of the holes/obstacles are functions of time, $\mathcal{H} = \mathcal{H}(t)$. Let $\Omega(t) = (P, \mathcal{H}(t))$. Let T_I and T_O be the *entry and exit time intervals*, during which the disks are allowed to enter and exit the domain. (Each of T_I and T_O is, in general, a finite set of intervals.) Assume that $\min\{t | t \in T_I\} = 0$, and let $T = \max\{t | t \in T_O\}$; the interval $[0, T]$ is called the *planning horizon*. A *path* or *trajectory* $\pi = \pi(t)$ is a continuous curve in the domain, $\pi : [t_I^\pi, t_O^\pi] \mapsto \Omega$, $t_I^\pi \in T_I$, $t_O^\pi \in T_O$, parameterized by time.

Definition 1. A path π is *feasible* if the unit disk whose center moves along π does not intersect any obstacle or the boundary of P (other than at the source or sink), and if the speed of motion along π is never greater than 1. A *collection* of feasible paths is *feasible* if for any two paths π, π' in the collection, the unit disks whose centers move along the paths do not intersect; i.e., $\forall t \in [t_I^\pi, t_O^\pi] \cap \left[t_I^{\pi'}, t_O^{\pi'}\right]$, $|\pi(t)\pi'(t)| \geq 2$.

Our objective is to find a feasible collection of source-sink paths of maximum cardinality. Let OPT_Ω denote an optimal collection, and let $|OPT_\Omega|$ be the number of paths in OPT_Ω. We will show how to find OPT_Ω paths for disks of radii somewhat smaller than 1 moving with speed somewhat greater than 1.

Hardness of the Problem

In the previous section we showed how to find, in polynomial time, a maximum number of thick paths in a domain with static obstacles. Note that we did not care about the path lengths, but only about the number of the paths. It was proved in [7] that finding a set of K thick paths, each with length smaller than a given bound L, is NP-hard (see [32] for details). The proof is by reduction from determining if there exists K length-bounded disjoint paths in a planar graph, which was shown NP-hard by Holst and Pina [36]. Vertex-disjoint paths in a plane graph correspond to disjoint thick paths in a polygonal domain, obtained by "fattening" the edges of the graph drawing.

Figure 9 reproduces an instance of the graph used by Holst and Pina [36] in their reduction from 3SAT. If we clip off the edges adjacent to S and r, fatten by 1 the remaining edges, and enclose the entire construction within a bounding rectangle, we transform the graph (drawing) into a polygonal domain Ω. Let the top and

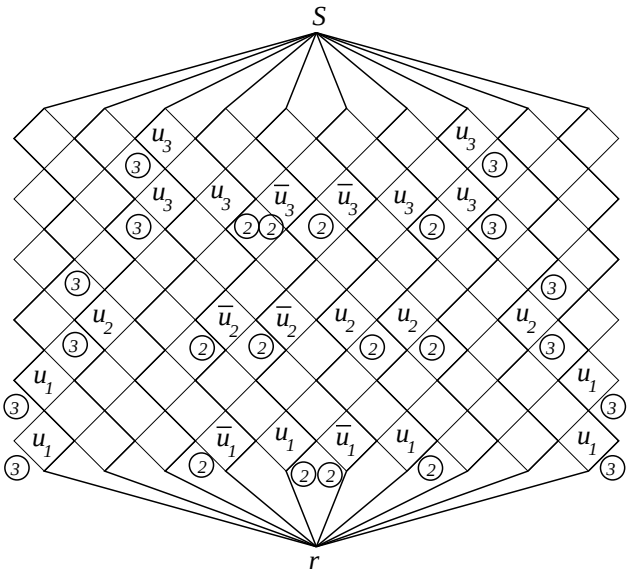

Figure 9: A reproduction of the instance in Figure 1 of [36]. Circled numbers show the edge weights; letters show the regions corresponding to some variables (u_1, u_2, u_3) in the 3SAT instance.

the bottom edges of the bounding rectangle be the source and the sink. Finally, let the entry and exit time intervals be just the single time interval $[0, L]$, where L is the length utilized in the NP-hardness proof of Theorem 1 of [36] (specifically, $L = 2n + 4$ for a set of n variables u_i). Then, it is clear that there exist K paths for unit disks in Ω if and only if there exist K disjoint paths of length at most L in the graph, which holds if and only if the 3SAT instance of [36] is satisfiable. We have:

Theorem 3.1. *Deciding whether $|OPT_\Omega| \geq K$ is NP-hard.*

Observe that the proof of the above theorem used *static* obstacles. If the obstacles are moving, finding just *one*, even *radius-0* bounded-speed path is NP-hard, as proved by Canny and Reif [1]. Note that the reduction in [1] uses some *fast* obstacles. We show below that under the assumption that the obstacles move with speed at most 1, it is possible to find $|OPT_\Omega|$ paths for disks with radius smaller than 1 and maximum speed greater than 1. In the remainder of this section we present our solution.

3.1 The Algorithm

Our algorithm has a very simple structure and is based on a uniform discretization of time. At each time slice, we consider a hexagonal disk packing in the free space. The disks from the packing form a graph in which each disk is connected to all disks at the next time slice that are reachable without intersecting the obstacles. We find a maximum number of paths in the graph and show that (possibly, after some local modifications) the paths correspond to a feasible collection of trajectories. The algorithm is summarized in Fig. 10.

To show correctness of the algorithm, we prove that, when lifted to the (x, y, t)-space, each feasible path contains a "stack" of cylinders. The consecutive cylinders in the stack overlap a lot by height and are only slightly shifted horizontally — this allows one to find a chain of *oblique* cylinders inside the stack; the bases of the oblique cylinders come from obstacle-free disks from the hexagonal packing. We search the graph, built from the oblique cylinders, for a maximum number of disjoint paths. Although the paths found may self-intersect, we show that "swapping" the paths locally and halving their radius resolves the intersections.

Lifting to (x, y, t). The domain Ω, $\Omega = \bigcup_{0 \leq t \leq T}(\Omega(t), t)$, is a 3D domain in the (x, y, t)-space. The holes, when moving, sweep a set $\mathcal{X} = \bigcup_{0 \leq t \leq T}(\mathcal{H}(t), t)$ of 3D obstacles. A feasible path π is a curve $(\pi(t), t)$,

Algorithm MaxThickPaths($\Omega(t), T_I, T_O$)
Input. Domain Ω with moving obstacles; entry and exit intervals T_I, T_O;
user-defined parameter $\Delta t \leq 1/3$.
Output. A feasible collection of $|OPT_\Omega|$ paths for
radius-$(\sqrt{3} - 1.5 + (2.25 - 1.5\sqrt{3})\Delta t)$ disks
moving with maximum speed $1.1/\Delta t - 0.8$.

$T \leftarrow \max\{T_O\} - \min\{T_I\}$ ▷ Planning horizon.
$M \leftarrow T/\Delta t$ ▷ It is assumed for simplicity that M is an integer.
$R \leftarrow 2(\sqrt{3} - 1.5 + (2.25 - 1.5\sqrt{3})\Delta t)$
$\mathcal{P}_{\bigcirc} \leftarrow$ hexagonal packing of R-disks
for $m = 0$ **to** M
$t_m \leftarrow m \cdot \Delta t$
$\mathcal{P}_m \leftarrow$ disks from $\mathcal{P}_{\bigcirc}$ free of obstacles at t_m
endfor
$\mathcal{P} \leftarrow \bigcup_m \mathcal{P}_m$
$G \leftarrow$ motion graph on $\mathcal{P}$
$\mathbf{f}(G) \leftarrow$ maximum number of disjoint paths in G
▷ Each path is a tube of radius R
Swap facing elementary cylinders used by $\mathbf{f}(G)$
Halve the radius of the tubes

Figure 10: Algorithm MaxThickPaths.

connecting a point $(\pi(t_I^\pi), t_I^\pi)$ in the rectangle $\Gamma_I \times T_I$ to a point $(\pi(t_O^\pi), t_O^\pi) \in \Gamma_O \times T_O$ (recall that we assume that P has Riemann sheets attached along Γ_I and Γ_O, from/through which the disks enter/exit Ω). The Minkowski sum of π with the (two-dimensional, horizontal) unit disk, $\langle\pi\rangle^{(1)}$, is a "slanted and curved" cylindrical *tube*; for a feasible collection of paths these tubes must not intersect the obstacles or each other. In what follows we will identify the paths with their (x, y, t) counterparts — the tubes.

Obstacles Motion. The condition that the obstacles "move with speed at most 1" formally means that $\mathcal{X}$ has a tangent plane almost everywhere, and that the plane is inclined by at least 45^o to the (x, y)-plane. We assume that the obstacles' motion is such that the following query can be answered in time polynomial in the input size: Given time t, two points $a, b \in \Omega$, speed v, and radius r, determine whether a disk of radius r intersects any obstacle when its center starts at point a at time t and moves at constant speed v along a straight line segment to point b. Apart from this, we impose no restrictions on the obstacles' motion; e.g., it is possible that during the motion the obstacles intersect, grow, disappear, etc., as long as the tangency condition is respected.

Naming Conventions. All cylinders in this paper have horizontal circular bases. By default, a "cylinder" means a *right* cylinder; oblique cylinders will be called *elementary* cylinders. By a *motion* or a *path* of a disk we understand a motion or a path of its center. The *horizontal distance* between two points is the distance between their projections on a horizontal plane. An r-disk is a disk of radius r.

Cylinders Inside a Feasible Path. The following lemma is straightforward:

Lemma 3.2. *Let C be an r-disk; let C' be C, shifted horizontally by $a < 2r$. Then the intersection $C \cap C'$ contains an $(r - a/2)$-disk.*

Let $\Delta t < 2/3$ be some constant. Let π be a feasible path. The next lemma shows that for any $t \in [t_I^\pi, t_O^\pi]$, there exists a "chunk" of obstacle-free space around $\pi(t)$.

Lemma 3.3. *The cylinder $\langle\pi(t)\rangle^{(1-\Delta t)} \times [t - \Delta t, t + \Delta t]$ is obstacle-free.*

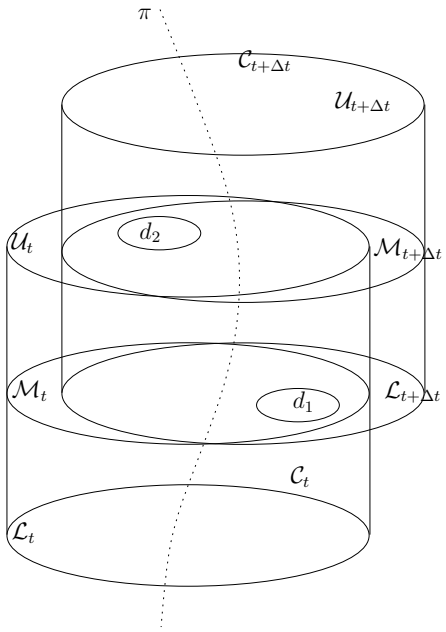

Figure 11: The cylinders inside $\langle\pi\rangle^{(1)}$. (The superscript "π" is omitted in the labels $\mathcal{L}_t^\pi, \mathcal{U}_t^\pi$, etc.).

Proof. At time t, the unit disk centered at $\pi(t)$ is obstacle-free. Since the obstacles' speed is bounded by 1, no obstacle could have been (resp., will be) closer than $1 - \Delta t$ to $\pi(t)$ during the time interval $[t - \Delta t, t]$ (resp., $[t, t + \Delta t]$). □

We denote the obstacle-free cylinder whose existence is established in the above lemma by $\mathcal{C}_t^\pi$. Let $\mathcal{L}_t^\pi$ (resp., $\mathcal{M}_t^\pi$, $\mathcal{U}_t^\pi$) be the cross-sections of $\mathcal{C}_t^\pi$ by the horizontal plane at $t - \Delta t$ (resp., t, $t + \Delta t$); i.e., $\mathcal{L}_t^\pi$ and $\mathcal{U}_t^\pi$ are the bases of $\mathcal{C}_t^\pi$. Refer to Fig. 11. (We assume that if $t - \Delta t < t_I^\pi$ (resp., $t + \Delta t > t_O^\pi$), the part of $\mathcal{C}_t^\pi$ below (resp., above) $\mathcal{M}_t^\pi$ is chopped off.)

It will be convenient to define a quantity R, dependent on Δt, as follows:

$$R = \frac{1 - \frac{3}{2}\Delta t}{1 + \frac{2}{\sqrt{3}}} \quad .$$

Lemma 3.4. *Inside the intersection $\mathcal{M}_t^\pi \cap \mathcal{L}_{t+\Delta t}^\pi$ there exists a $(1 + 2/\sqrt{3})R$-disk.*

Proof. Since the speed of motion along a feasible path is at most 1, $\mathcal{L}_{t+\Delta t}^\pi$ is $\mathcal{M}_t^\pi$ shifted horizontally by at most Δt. The lemma follows now from Lemma 3.2. □

Let $d_1 \subset (\mathcal{M}_t^\pi \cap \mathcal{L}_{t+\Delta t}^\pi)$ (resp., $d_2 \subset (\mathcal{M}_{t+\Delta t}^\pi \cap \mathcal{U}_t^\pi)$) be an R-disk, fully lying inside the intersection of the cross-sections (Fig. 11); let c_1 (resp., c_2) be the center of the disk.

Lemma 3.5. *The horizontal distance between c_1 and c_2 is at most D, where*

$$D = 2\sqrt{(1 - \Delta t - R)^2 - \left(\frac{\Delta t}{2}\right)^2} \quad .$$

Proof. Let d_2' be the projection of d_2 onto the plane of d_1. The maximum of the distance between d_1 and d_2' is attained when the disks lie at opposite "ends" of the lune $\mathcal{M}_t^\pi \cap \mathcal{L}_{t+\Delta t}^\pi$ (Fig. 12). The maximum is equal to D. □

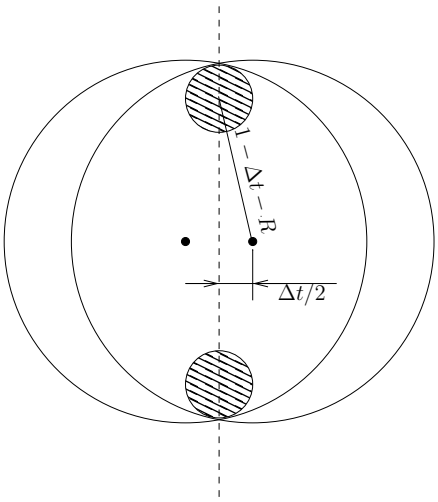

Figure 12: The dots are the centers of $\mathcal{M}_t^\pi$ and $\mathcal{L}_{t+\Delta t}^\pi$ — disks of radius $1-\Delta t$, shifted by at most Δt. The hatched disks are d_1 and d_2' in the position when the distance between them is maximized.

The quantities R and D introduced above will be used frequently in the rest of the analysis. The specific choice of R (and of Δt) will be justified in the sequel.

Lemma 3.6. *The R-disk d_1 can be moved to d_2 by a straight-line motion with speed at most $D/\Delta t$ without intersecting any obstacle.*

Proof. Connect d_1 to d_2 by the tube $\langle c_1 c_2 \rangle^{(R)}$ — the convex hull of the disks d_1 and d_2. Since both disks lie inside the (obstacle-free) cylinder $\mathcal{C}_t^\pi$, the tube does not intersect any obstacle. By Lemma 3.5, the speed of motion along c_1c_2 is at most $D/\Delta t$. □

Slicing the Time. Assume for simplicity that $T/\Delta t$ is an integer, M. Partition the planning horizon, $[0,T]$, into M intervals each of length Δt. Let $t_0 \dots t_M$ denote the intervals' endpoints; $t_m = m\Delta t, m \in \{0 \dots M\}$. Applying Lemmas 3.4 and 3.6 at each interval, we get:

Corollary 3.7. *Inside a feasible tube $\langle \pi \rangle^{(1)}$ there exists a set of cylinders $\{\mathcal{C}_m^\pi\}_{m=0}^M$ with the following properties:*

1: *the intersection $\mathcal{M}_m^\pi \cap \mathcal{L}_{m+1}^\pi$ contains a $(1+2/\sqrt{3})R$-disk and, hence, so does $\mathcal{M}_{m+1}^\pi \cap \mathcal{U}_m^\pi$; and,*

2: *for any R-disks $d_1 \subset (\mathcal{M}_m^\pi \cap \mathcal{L}_{m+1}^\pi)$ and $d_2 \subset (\mathcal{M}_{m+1}^\pi \cap \mathcal{U}_m^\pi)$ there exists a straight-line motion, with speed at most $D/\Delta t$, that moves d_1 to d_2 without intersecting any obstacle.*

Disk Packing. Let $\mathcal{P}_{\bigcirc}$ be an (infinite) hexagonal packing of R-disks in the plane. Let $\mathcal{P}_m$, for $m = 0 \dots M$, be the set of disks within $\Omega(t_m)$ obtained by removing from $\mathcal{P}_{\bigcirc}$ all disks intersected by $\mathcal{H}(t_m)$ or by P. Let $\mathcal{P} = \cup_m \mathcal{P}_m$.

Remark 3. $\mathcal{P}_m$ is not necessarily a *maximal* packing. The obstacles at t_m cause some disks from $\mathcal{P}_{\bigcirc}$ to be removed, after which there may exist free space to add more disks to $\mathcal{P}_m$. Although it cannot hurt to augment $\mathcal{P}_m$ into a maximal packing by adding such disks, we ignore this possibility in our analysis.

Motion Graph. Let $G = (\mathcal{P}, E)$ be the *motion graph* — a directed graph with vertices corresponding to the disks in the packings, and edges defined as follows. For disks $d_1, d_2 \in \mathcal{P}$ there is an edge $(d_1, d_2) \in E$ whenever the vertical distance between d_1 and d_2 is Δt (i.e., $d_1 \in \mathcal{P}_m$, $d_2 \in \mathcal{P}_{m+1}$ for some m), *and* there exists a straight-line motion, with speed at most $D/\Delta t$, that moves d_1 to d_2 without intersecting any obstacle (Fig. 13). Add a super-source vertex S (resp., super-sink vertex T) to $\mathcal{P}$; connect S (resp., T) to the disks

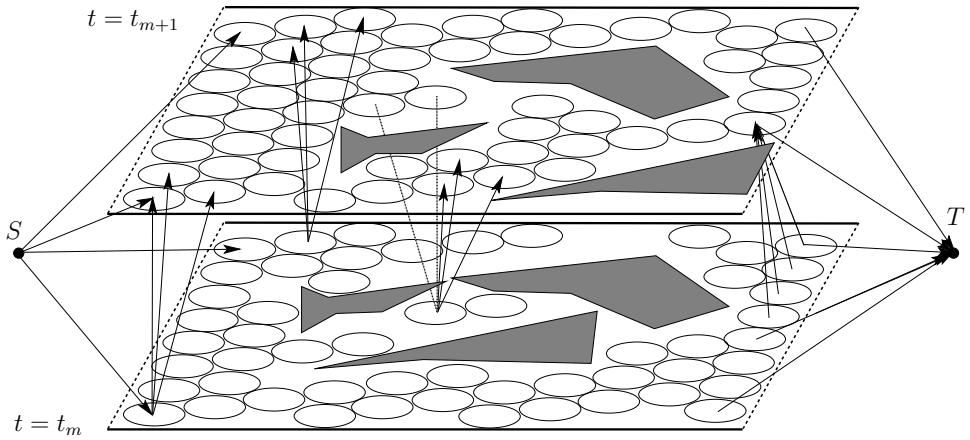

Figure 13: P is a rectangle; Γ_I, Γ_O are dashed. The edges of G connect each disk to all other disks that it can reach within time Δt without intersecting obstacles. Some of the edges are shown. The dotted segments are not edges because their elementary cylinders are intersected by an obstacle path.

whose centers are closer than $2R$ to Γ_I (resp., Γ_O) for $t_m \in T_I$ (resp., $t_m \in T_O$). By a *path in* G we will always mean an S-T path.

We will assume that G is embedded in (x, y, t) space with vertices at the centers of the disks. Depending on the context, by the *edge* e of G we will mean four related things: (1) the directed edge of the graph; (2) the vector, directed upward, in the (x, y, t)-space; (3) the (undirected) segment in the (x, y, t)-space; and, (4) the oblique cylinder $\langle e \rangle^{(R)}$, which we call an *elementary cylinder*. The specific meaning will be clear from the context; e.g., in the expression $\langle e \rangle^{(r)}$ for $r \neq R$, the edge e will necessarily mean a segment (meaning (3)). A path in G will be identified with the tube, obtained by inflating the edges of the path (excluding the edges adjacent to S and T).

By construction, any path π^G in G is an obstacle-free path for an R-disk; $\langle \pi^G \rangle^{(R)} \cap \mathcal{X} = \emptyset$. In the next lemma we show that, conversely, there also exists a path in G for each feasible thickness-1 path in Ω.

Lemma 3.8. *Let π be a feasible path in Ω. There exists a path π^G in G such that $\langle \pi^G \rangle^{(R)} \subset \langle \pi \rangle^{(1)}$.*

Proof. Any obstacle-free $(1+2/\sqrt{3})R$-disk d in $\Omega(t_m)$ fully contains at least one disk from $\mathcal{P}_m$. Indeed, let c be the center of d. For any point in the plane (in particular, for c) there exists a disk d' in $\mathcal{P}_{\bigcirc}$ whose center c' lies within $2/\sqrt{3}R$ from c. Thus, the distance from c' to the boundary of d is $(1 + 2/\sqrt{3})R - |cc'| \geq R$, which means that $d' \subset d$. Moreover, since d is obstacle-free, d' is also obstacle-free. Hence d' "survives" in $\mathcal{P}_m$: $d' \in \mathcal{P}_m$.

By Corollary 3.7, property 1, each of $\mathcal{M}^\pi_m \cap \mathcal{L}^\pi_{m+1}$, $\mathcal{M}^\pi_{m+1} \cap \mathcal{U}^\pi_m$ contains a $(1+2/\sqrt{3})R$-disk. Thus, there exists a disk d_1 from $\mathcal{P}_m$ inside $\mathcal{M}^\pi_m \cap \mathcal{L}^\pi_{m+1}$, and a disk $d_2 \in \mathcal{P}_{m+1}$ inside $\mathcal{M}^\pi_{m+1} \cap \mathcal{U}^\pi_m$. By Corollary 3.7, property 2, $(d_1, d_2) \in E$. □

MaxFlow in G. In the sequel, whenever we speak of a *flow* we will mean an *integral* S-T flow in G. Assign capacity 1 to each vertex and each edge of G. Let $\mathbf{f}(G)$ denote the maximum flow; let $|\mathbf{f}(G)|$ be its value. By the Flow Decomposition Theorem, $\mathbf{f}(G)$ can be decomposed into a set of $|\mathbf{f}(G)|$ vertex-disjoint paths. By Lemma 3.8, $|\mathbf{f}(G)| \geq |OPT_\Omega|$. By construction, none of the $|\mathbf{f}(G)|$ paths intersects an obstacle. Also, at any given time slice, the paths are pairwise-disjoint since they go through disjoint disks in the packing. Nevertheless, nothing prevents the paths from intersecting each other *between* the time slices. Next we show how to decrease the radius of the paths and to "untangle" them to guarantee that such intersections (if any) are resolved.

The Value of Δt. As shown below, choosing $\Delta t \leq 1/3$, ensures that the maximum degree of G is 7, which is as small as possible. The quest for low degree is prompted by the following. To quantify the decrease of

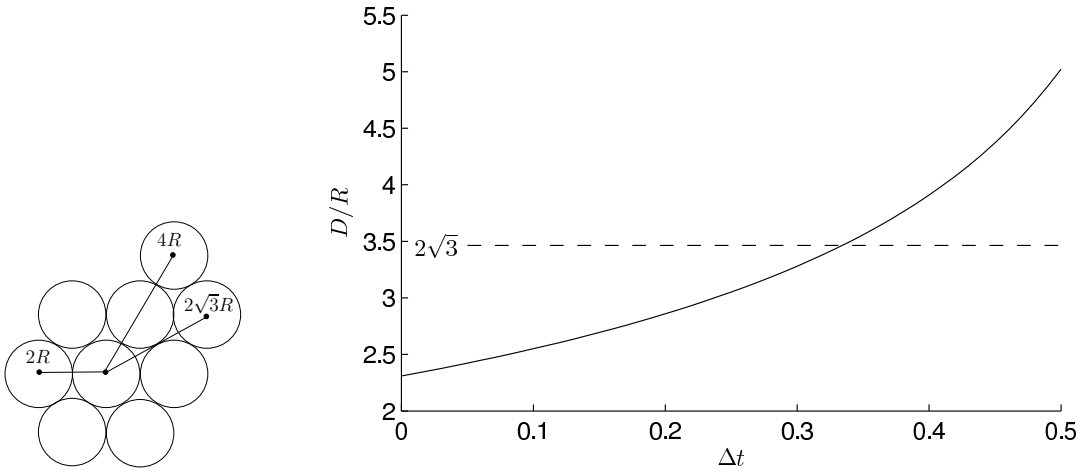

Figure 14: The degree of a disk changes at certain values of D/R, whose dependence on Δt is shown on the right.

the tube radius, necessary to resolve the intersections of disks that can occur between the time slices, we analyze *all possible* intersections between elementary cylinders. The higher the maximum possible degree of a vertex in G, the more complex the analysis is. Although we believe that our algorithm works for other choices of Δt as well, its analysis and proof of correctness become quite complicated. Also, for larger Δt the resulting radius of the disks becomes much smaller, unless more elaborate uncrossings of the cylinders (involving, say, "bending" of the tubes) is performed.

By construction, a disk is connected to all disks at the next time slice whose centers lie within horizontal distance D from the disk's center. Thus, the maximum degree of a disk in G depends on the ratio of D to R. If $D/R < 2\sqrt{3}$, any disk can potentially be connected only to the disk directly above and to its 6 neighbors; if $D/R < 4$, the disk can be connected to at most 6 more disks, if $D/R < 2\sqrt{7}$ — to 6 more, and so on (Fig. 14, left). The ratio D/R, in turn, depends on Δt, as shown in Figure 14, right. It is easy to check that with $\Delta t \leq 1/3$, we have $D/R < 2\sqrt{3}$, so that the maximum degree of a disk in the motion graph is 7.

Intersection of Elementary Cylinders. Let $e_1 = (a_1, b_1)$, $e_2 = (a_2, b_2)$, $e_1, e_2 \in E$, be two elementary cylinders from one layer of G. Let p_i be a point moving along e_i, for $i = 1, 2$; let $p_i(\tau)$ be the position of p_i when it is at distance τ from the plane containing the lower bases of e_1, e_2.

Definition 2. The *relative displacement* of e_1, e_2, denoted $\mathsf{d}(e_1, e_2)$, is the minimum distance between the cross-sections of the axes of e_1, e_2 by a horizontal plane: $\mathsf{d}(e_1, e_2) = \min_\tau |p_1(\tau)p_2(\tau)|$. The *height*, $\mathsf{h}(e_1, e_2)$, is the value of τ at which the minimum is attained.

Let $\mathbf{l} = a_2 - a_1, \mathbf{u} = b_2 - b_1$ (Fig. 15). We have:

$$p_2(\tau) - p_1(\tau) = \mathbf{l} + \frac{\tau}{\Delta t}(\mathbf{u} - \mathbf{l}) \quad \Longrightarrow$$

$$|p_1(\tau)p_2(\tau)|^2 = \frac{\tau^2}{\Delta t^2}(\mathbf{u} - \mathbf{l})^2 + 2\frac{\tau}{\Delta t}\mathbf{l} \cdot (\mathbf{u} - \mathbf{l}) + |\mathbf{l}|^2 \quad \Longrightarrow$$

$$\mathsf{h}(e_1, e_2) = \frac{\mathbf{l} \cdot (\mathbf{l} - \mathbf{u})}{|\mathbf{l} - \mathbf{u}|^2}\Delta t \quad , \tag{1}$$

$$[\mathsf{d}(e_1, e_2)]^2 = |\mathbf{l}|^2 - \frac{(\mathbf{l} \cdot (\mathbf{u} - \mathbf{l}))^2}{(\mathbf{u} - \mathbf{l})^2} = \frac{|\mathbf{l} \times (\mathbf{u} - \mathbf{l})|^2}{|\mathbf{u} - \mathbf{l}|^2} \quad \Longrightarrow$$

$$\mathsf{d}(e_1, e_2) = \frac{|\mathbf{l} \times \mathbf{u}|}{|\mathbf{u} - \mathbf{l}|} \quad . \tag{2}$$

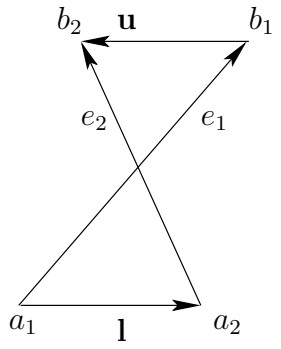

Figure 15: a_1b_1, a_2b_2 are the axes of the cylinders.

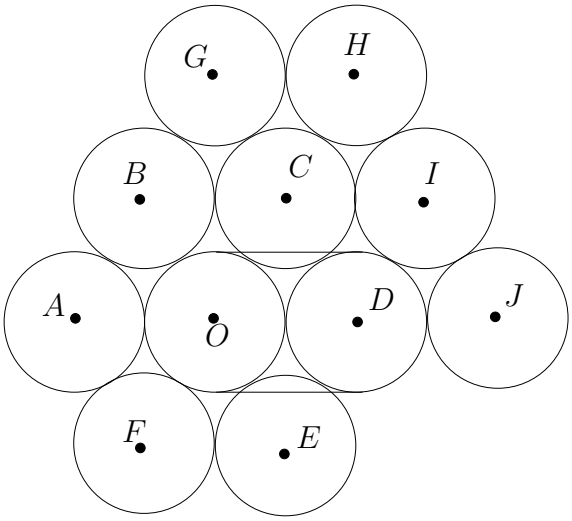

Figure 16: We consider intersection with the cylinders whose lower bases are centered at O.

In what follows we show that the relative displacement of a pair of intersecting elementary cylinders is either 0 (in which case we "swap" the cylinders) or is greater than R (so that decreasing the radius of the tubes to $R/2$ removes the intersections). We consider all possible pairs of intersecting cylinders. By symmetry, it is enough to look at (the 7) cylinders with common lower base centered at a point O at some time slice t_m (Fig. 16). For any disk center C at t_m, denote by C' the point at t_{m+1} directly above C (C' is the center of the corresponding disk from $\mathcal{P}_{m+1}$).

A Vertical Cylinder. The cylinder OO' may be intersected by 12 cylinders (AB', BA', BC', CB', CD', DC', DE', ED', EF', FE', FA', AF'). Without loss of generality, consider the intersection of OO' and AB'. In this case, $\mathbf{l} = (-2R, 0)$ and $\mathbf{u} = (-R, \sqrt{3}R)$, so

$$d(OO', AB') = \frac{|\mathbf{l} \times \mathbf{u}|}{|\mathbf{u} - \mathbf{l}|} = \sqrt{3}R \quad .$$

Swapping "Facing" Oblique Cylinders. Consider the cylinders OA' and AO'. Since their axes intersect, no decrease of their radius could possibly resolve their intersection. In case the maxflow in G uses a pair of such cylinders, we simply "swap" them, routing the flow via OO' and AA'. Since OO' is vertical, its relative displacement with any intersecting cylinder equals $\sqrt{3}R$. The same is true for AA'.

General Oblique Cylinders. To establish what a minimum *positive* relative displacement between a pair of intersecting oblique cylinders is, consider the cylinder OD'. By symmetry it is enough to look only at the cylinders whose bases' centers project onto the plane of Figure 16 *on or above* the line OD (for instance, we will not consider cylinders with bases centered at E or F).

Consider the cylinders, with bases disjoint from the bases of OD', whose projections intersect with the projection of OD' — only these cylinders may possibly intersect OD'. There are 18 such cylinders (apart from DO', which faces OD'): AO', BO', CO', AB', BA', BC', CB', CG', GC', CH', HC', CI', IC', DC', DI'. Of these, AO', BC', CI', DJ' have axes parallel to OD'. Since their bases are disjoint from the bases of OD', none of them intersects OD'. For the remaining 14 cylinders,

$$BO', CO', AB', BA', CB', CG', GC', CH', HC', IC', DC', DI', IJ', JI', \tag{3}$$

we calculate the relative displacement with OD' explicitly.

To make the calculation of the relative displacements easier, we rewrite the formulae for $\mathbf{l}$ and $\mathbf{u}$ in terms of the (integer) coordinates of the cylinders bases centers in the "triangular" coordinate system with the origin at O, first axis horizontal, and the other axis inclined by 60^o to the horizontal axis; the unit along the axes will be $2R$. In this coordinate system the coordinates of the points are:

$$O(0,0)\,,\ A(-1,0)\,,\ B(-1,1)\,,\ C(0,1)\,,\ D(1,0)\,,\ G(-1,2)\,,H(0,2)\,,\ I(1,1)\,,\ J(2,0). \tag{4}$$

In general, for points $X(i,j)$, $Y(k,l)$, we have in the Cartesian coordinate system $X = 2R(i + j/2, \frac{\sqrt{3}}{2}j)$, $Y = 2R(k + l/2, \frac{\sqrt{3}}{2}l)$. To calculate the relative displacement of the cylinder XY' with OD' we use

$$\mathbf{l} = XO = 2R(i + j/2, \frac{\sqrt{3}}{2}j)\,,\ \mathbf{u} = Y'D' = 2R(k + l/2 - 1, \frac{\sqrt{3}}{2}l). \tag{5}$$

We wrote a simple MATLAB script[1] that calculates the relative displacement of each of the cylinders in (3) with OD'; the script uses the formulae (1), (2), (5) and the list (4). The calculated relative displacements and their heights are presented in Table 1 (of course, for two cylinders, e_1, e_2, to intersect at all, the height $h(e_1, e_2)$ must be between 0 and Δt). The smallest relative displacement of OD' is with CO' and DC'; the relative displacement equals R (which can also be easily seen from Fig. 16 as the distance between the midpoint of OD and midpoints of CO and DC is R).

Putting Things Together. It follows that after the maxflow in G is found, all intersections between the elementary cylinders will be resolved after the facing cylinders, used by the flow, are swapped and the radius of the tubes is halved. The algorithm MaxThickPaths in Fig. 10 implements the steps described above. Our main result is:

Theorem 3.9. *For any $\Delta t \le 1/3$, algorithm* MaxThickPaths *computes a collection of (at least) $|OPT_\Omega|$ feasible paths for radius-$(\sqrt{3} - 1.5 + (2.25 - 1.5\sqrt{3})\Delta t)$ disks, moving with speed at most $1.1/\Delta t - 0.8$. The algorithm runs in time polynomial in n, $T/\Delta t$ and N, where N is the value of largest coordinate of Ω.*

Proof. The correctness of the algorithm follows from the above discussion; the running time follows from a straightforward implementation of the steps. There are $O(N^2)$ disks intersecting P. At each time slice, checking whether a disk is obstacle-free can be done in $O(n)$ time. Since the maximum degree of the motion graph is constant (7), the maximum number of edges in the graph, between two consecutive time slices, is

[1] Our MATLAB codes for generating Table 1 and Fig. 14, right are available at `http://www.cs.helsinki.fi/valentin.polishchuk/pages/penetr.m` and `http://www.cs.helsinki.fi/valentin.polishchuk/pages/DtoR.m`. The latter code is also used to check that for $\Delta t \le 1/3$ we have

$$\frac{D}{\Delta t} < 1.1/\Delta t - 0.8,$$

where the right-hand side is just a more comprehensible expression, for the maximum speed of disks, than the exact one — $2\sqrt{(1 - \Delta t - R)^2 - \left(\frac{\Delta t}{2}\right)^2}/\Delta t$.

Cylinder	$h(\cdot, OD')/\Delta t$	$d(\cdot, OD')/2R$
BO'	0.5000	0.8660
CO'	0.5000	0.5000
AB'	-0.5000	0.8660
BA'	-0.0000	1.0000
CB'	0.2500	0.8660
CG'	0.0000	1.0000
GC'	1.5000	0.8660
CH'	-0.5000	0.8660
HC'	1.0000	1.0000
IC'	0.7500	0.8660
DC'	0.5000	0.5000
DI'	0.5000	0.8660
IJ'	1.5000	0.8660
JI'	1.0000	1.0000

Table 1: The relative displacements with OD'.

linear in the maximum number of disks at a time slice, i.e., $O(N^2)$. Testing whether an edge is in the motion graph can be done in polynomial time by the assumption about the obstacles' motion (page 11). Overall, the motion graph has $O(N^2T/\Delta t)$ vertices and edges; thus, maximum flow in it can be found in time polynomial in N and $T/\Delta t$ [4]. Each edge used by the flow faces only one edge; swapping the edges, if needed, takes constant time. □

4 Conclusion

Several directions remain open for future research.

For our motivating application in ATM it may be useful to compute a realizing set of trajectories (in (x, y, z, t)-space).

When snapped onto the disks from the hexagonal packing, even a straight path will yield a path with many bends, one at each time slice. For our motivating ATM application, it would be interesting to solve the problem in which routes are restricted to have a bounded number of turns.

We assumed that the maximum speed of the obstacles' motion equals the speed limit of the disks. In our ATM application, the moving obstacles may correspond to weather systems, which move at speeds much lower than the aircraft that avoid them. This suggests that we investigate the case in which the obstacles move at a speed much different from the disks and determine how the approximation guarantees of our algorithm vary with the bounds on obstacle speed.

We leave open the problem of finding a thick "wire" (a non-self-overlapping thick path) in a polygonal domain with static obstacles. While finding a thick path is straightforward (just offset the obstacles by 1 and search for a thin path in the new domain), how does one compute a thick path that is non-self-overlapping (or determine that none exists)?

We conjecture that our approach can be extended to higher dimensions and to other shapes of the moving objects (as long as the motion is purely translational).

Finally, it would be interesting to implement our algorithms and see how well they perform in practice. In particular, how fast do the disks move through the domain along the paths computed by our algorithm?

Acknowledgements

We thank the anonymous referees and Irina Kostitsyna for many helpful suggestions and corrections. E. Arkin is partially supported by NSF (CCF-0431030, CCF-0729019). J. Mitchell is partially supported by NSF

(CCF-0431030, CCF-0528209, CCF-0729019), NASA Ames, and Metron Aviation. V. Polishchuk is supported in part by Academy of Finland grant 118653 (ALGODAN).

References

[1] J. Canny, J. H. Reif, New lower bound techniques for robot motion planning problems, in: Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci., 1987, pp. 49–60.

[2] J. S. B. Mitchell, Geometric shortest paths and network optimization, in: J.-R. Sack, J. Urrutia (Eds.), Handbook of Computational Geometry, Elsevier Science B.V. North-Holland, Amsterdam, 2000, pp. 633–701.

[3] M. R. Silver, O. L. de Weck, Time-expanded decision networks: A framework for designing evolvable complex systems, Syst. Eng. 10 (2) (2007) 167–188. doi:http://dx.doi.org/10.1002/sys.v10:2.

[4] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Englewood Cliffs, NJ, 1993.

[5] G. Strang, Maximal flow through a domain, Math. Program. 26 (1983) 123–143.

[6] J. S. B. Mitchell, On maximum flows in polyhedral domains, J. Comput. Syst. Sci. 40 (1990) 88–123.

[7] J. S. B. Mitchell, V. Polishchuk, Thick non-crossing paths and minimum-cost flows in polygonal domains, in: 23rd ACM Symposium on Computational Geometry, 2007, pp. 56–65.

[8] J. Krozel, J. S. B. Mitchell, V. Polishchuk, J. Prete, Capacity estimation for level flight with convective weather constraints, Air Traffic Control Quarterly 15 (3) (2007) 209–238.

[9] J. Krozel, J. S. B. Mitchell, V. Polishchuk, J. Prete, Airspace capacity estimation with convective weather constraints, in: AIAA Guidance, Navigation, and Control Conference, 2007.

[10] J. S. B. Mitchell, V. Polishchuk, J. Krozel, Airspace throughput analysis considering stochastic weather, in: AIAA Guidance, Navigation, and Control Conference, 2006.

[11] N. Aggarwal, K. Fujimura, Motion planning amidst planar moving obstacles, in: Proceedings of ICRA, 1994, pp. 2153–2158.

[12] J. Canny, A. Rege, J. Reif, An exact algorithm for kinodynamic planning in the plane, Discrete Comput. Geom. 6 (1991) 461–484.

[13] K. Fujimura, Motion planning amid transient obstacles, The International Journal of Robotics Research 13 (5) (1994) 395–407.

[14] K. Fujimura, Time-minimal paths amidst moving obstacles in three dimensions, Theor. Comput. Sci. 270 (1-2) (2002) 421–440.

[15] K. Fujimura, H. Samet, Planning a time-minimal motion among moving obstacles, Algorithmica 10 (1) (1993) 41–63.

[16] J. Reif, M. Sharir, Motion planning in the presence of moving obstacles, J. ACM 41 (4) (1994) 764–790.

[17] J. P. van den Berg, D. Ferguson, J. Kuffner, Anytime path planning and replanning in dynamic environments, in: ICRA, 2006, pp. 2366–2371.

[18] J. van den Berg, Path planning in dynamic environments, Ph.D. thesis, Utrecht University (2007).

[19] J. P. van den Berg, M. H. Overmars, Roadmap-based motion planning in dynamic environments, IEEE Transactions on Robotics 21 (5) (2005) 885–897.

[20] J. P. van den Berg, M. H. Overmars, Kinodynamic motion planning on roadmaps in dynamic environments, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS'07, 2007, pp. 4253 – 4258.

[21] D. Nieuwenhuisen, J. P. van den Berg, M. H. Overmars, Efficient path planning in changing environments, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS'07, 2007, pp. 3295–3301.

[22] J. P. van den Berg, M. H. Overmars, Planning the shortest safe path amidst unpredictably moving obstacles, in: S. Akella, N. M. Amato, W. H. Huang, B. Mishra (Eds.), Proc. Workshop on Algorithmic Foundations of Robotics - WAFR'06, Vol. 47 of Springer Tracts in Advanced Robotics, Springer, 2008, pp. 103–118.

[23] J. P. van den Berg, D. Nieuwenhuisen, L. Jaillet, M. H. Overmars, Creating robust roadmaps for motion planning in changing environments, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS'05, 2005, pp. 2415–2421.

[24] J.-C. Latombe, Robot Motion Planning, Kluwer, Boston, 1991.

[25] Z. Li, J. F. Canny (Eds.), Nonholonomic Motion Planning, Kluwer, Norwell, MA, 1992.

[26] J. S. B. Mitchell, Shortest paths and networks, in: J. E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, 1997, Ch. 24, pp. 445–466.

[27] A. Dumitrescu, I. Suzuki, P. Zylinski, Offline variants of the "lion and man" problem, in: Symposium on Computational Geometry, 2007, pp. 102–111.

[28] J. P. van den Berg, M. H. Overmars, Prioritized motion planning for multiple robots, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS'05, 2005, pp. 2217–2222.

[29] J. Prete, Aircraft routing in the presence of hazardous weather, Ph.D. thesis, Stony Brook University (Aug 2007).

[30] J. Krozel, S. Penny, J. Prete, J. S. B. Mitchell, Comparison of algorithms for synthesizing weather avoidance routes in transition airspace, in: AIAA Guidance, Navigation and Control Conference, 2004.

[31] J. Prete, J. S. B. Mitchell, Safe routing of multiple aircraft flows in the presence of time-varying weather data, in: AIAA Guidance, Navigation and Control Conference, 2004.

[32] V. Polishchuk, Thick non-crossing paths and minimum-cost continuous flows in geometric domains, Ph.D. thesis, Stony Brook University, available at `http://cs.helsinki.fi/valentin.polishchuk/pages/thesis.pdf` (Aug 2007).

[33] L. Gewali, A. Meng, J. S. B. Mitchell, S. Ntafos, Path planning in $0/1/\infty$ weighted regions with applications, ORSA J. Comput. 2 (3) (1990) 253–272.

[34] T. Hu, A. Kahng, G. Robins, Optimal robust path planning in general environments, IEEE Transactions on Robotics and Automation 9 (1993) 775–784.

[35] J. Hershberger, S. Suri, An optimal algorithm for Euclidean shortest paths in the plane, SIAM J. Comp. 28 (1999) 2215–2256.

[36] H. v. d. Holst, J. C. d. Pina, Length-bounded disjoint paths in planar graphs, Discr. Appl. Math. 120 (1-3) (2002) 251–261. http://dx.doi.org/http://dx.doi.org/10.1016/S0166-218X(01)00294-3 `doi:http://dx.doi.org/10.1016/S0166-218X(01)00294-3`.