Motion Planning of Disk and Rectangular Robots

Eduard Eiben* Iyad Kanj† Salman Parsa‡

Abstract

We study the parameterized complexity of the motion planning of geometric-shaped robots in the plane, where the task is to move $k \in \mathbb{N}$ robots from their starting to their destination points without collisions.

We focus on disk and axis-aligned rectangular robots and on rectilinear motion. We consider both types of motions: coordinated (i.e., robots move in parallel) and serial. We also consider two objective functions: the total rectilinear length traveled and the total number of moves.

We prove that, in the presence of (rectangular) obstacles, the coordinated rectilinear motion planning of rectangular robots with the target of minimizing the total number of moves, is W[1]-hard parameterized by the number k of robots. This gives strong evidence that the problem is unlikely to be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$, for some computable function f, where n is the number of bits that encode the coordinates of the robots and obstacles, contrasting the recent result for the setting of obstacle-free motion in the plane, which showed the problem to be fixed-parameter tractable (FPT).

We then focus on the case of free motion in the plane, and consider the rectilinear motion of axis-aligned rectangular robots, with the goal of minimizing their total rectilinear travel length. We show that the restriction in which we require each robot to travel along a shortest rectilinear path, is NP-hard.

Finally, we consider the free rectilinear motion of congruent disk robots in the plane.

We show that the problem is FPT parameterized by k, for both target functions, if we restrict the motion of the centers of the disks to a unit grid, a restriction that has applications in real-world robotic problems.

1 Introduction

Motivation and Related Work. We investigate the parameterized complexity of several motion planning problems of disk robots and of axis-aligned rectangular robots in the plane, where the parameter under consideration is the number of robots. In this setting, we are given a set of k robots, each with a starting and ending position, and the goal is to compute a schedule in which each robot reaches its destination without colliding with other robots, and while minimizing a certain objective function. The two objective functions under consideration are the total rectilinear length traveled by all the robots in the schedule, and the total number of moves in the schedule. We consider only rectilinear translation motion, that is, each move is a translation along either a horizontal or vertical direction. We consider two settings: the setting of free motion in the plane (i.e., motion in the plane with no obstacles), and the setting where rectangular obstacles may be present. We also consider two types of motion: coordinated, where robots may move in parallel, and serial, where robots move one at a time. Most of the problems under consideration are either NP-hard or PSPACE-hard.

Our work assumes a Turing machine model and that the input length is the number of bits needed to represent the coordinates of the points defining the geometric objects (i.e., robots/obstacles) in the problem instance. Note that this number can be much larger than the parameter k. We believe that this model is more realistic to the problems under consideration than the real RAM model [35], which assumes that arithmetic operations over the reals can be performed in constant time.

There has been an enormous body of work on geometric motion planning problems, mainly focusing on the feasibility (i.e., whether any schedule exists), dating back to the works of Schwartz and Sharir in the 1980s [32–34]. They showed that deciding the feasibility of a problem instance for two disks in a region bounded by n "walls" can be done in time $\mathcal{O}(n^3)$ [32]; their result can be generalized to any number, k, of disks to yield an $\mathcal{O}(n^{h(k)})$-time algorithm, for some function h of k. Ramanathan and Alagar [30] improved this to $\mathcal{O}(n^k)$, conjecturing that this running time is asymptotically optimal. The feasibility of the coordinated motion planning of rectangular robots confined to a bounding box was shown to be PSPACE-hard [22, 23], even for congruent square robots [38]. The problem of moving disks among polygonal obstacles in the plane was shown to be strongly NP-hard [28].

More recently, there has been quite some work on the continuous collision-free motion of a constant number of rectangles in the plane, to optimize or approximate the total Euclidean traveled length; we refer to [2, 15, 27] for

*Department of Computer Science, Royal Holloway, University of London, Egham, UK, eduard.eiben@rhul.ac.uk

†School of Computing, DePaul University, Chicago, USA, ikanj@cdm.depaul.edu

‡School of Computing, DePaul University, Chicago, USA, s.parsa@depaul.edu

some of the most recent works on this topic.

Dumitrescu and Jiang [12] studied the problem of moving unit disks in an obstacle-free environment. They consider two types of motion: translation and sliding (a move along a continuous curve). In a single step, a unit disk may move any distance either along a line (translation) or a curve (sliding) provided that it does not collide with another disk. They showed that deciding whether the disks can reach their destinations within $\ell \in \mathbb{N}$ moves is NP-hard, for either of the two motion types. For more information, we refer to the survey [11].

Recently, there has been a surge of works on the parametrized complexity of combinatorial variants of coordinated motion planning problems (i.e., on grids and graphs) [7,8,13,14,17,18]. Notably, minimizing the total number of moves and the total travel length (referred to as the energy) for the classical coordinated motion planning problem on grids featured as the Symposium on Computational Geometry (SoCG 2021) Challenge problem [16], due to its applications in artificial intelligence [4,36,39] and robotics [3,21,37].

Very recently, the parameterized complexity of translating axis-aligned rectangles in the free plane, with the goal of minimizing the number of moves was studied by Kanj and Parsa [24]. It was shown in [24] that the problem is FPT parameterized by the number of robots. The parameterized complexity for other shapes – in particular congruent disks – and for environments with obstacles were posed as open problems in [24]. Moreover, the work in [24] focused on the total number of moves, and did not consider the variants of the problems in which the goal is to minimize the travel length/energy.

In this paper, we study the parameterized complexity of these fundamental geometric motion planning problems, answering some of the open questions in [24], and extending their work to the objective function of minimizing the total travel length.

Contributions. We present both hardness and algorithmic results for the motion planning problems of disk robots and of axis-aligned rectangular robots in the plane. Our main contributions are:

(i) We prove that, in the presence of rectangular obstacles, the coordinated rectilinear motion planning of rectangular axis-aligned robots with the target of minimizing the total number of moves, is W[1]-hard parameterized by the number k of robots, giving strong evidence that it is unlikely to be solved in time $f(k)\cdot n^{\mathcal{O}(1)}$, for some computable function f. This answers an open question in [24], and contrasts the case of motion in the free plane, where the problem is FPT [24].

We then consider free motion and show:

(ii) The restriction of the rectilinear motion of axis-aligned rectangular robots, with the goal of minimizing the rectilinear total travel length, to instances in which each robot is required to travel along a shortest rectilinear path (i.e., whose length is equal to the rectilinear distance between the starting and ending positions of the robot), is NP-hard. A byproduct of this hardness result – among others that we show in this paper – is the NP-hardness of a restriction of the classical coordinated motion planning problem on full rectangular grids, to instances where we require each robot to travel along a shortest grid path between its starting and ending position (i.e., whose length is equal to the Manhattan distance between the starting and ending position of the robot on the grid). We also describe an FPT algorithm for the rectilinear motion of axis-aligned rectangular robots, with the goal of minimizing the rectilinear total travel length.

Finally, we consider the rectilinear free motion of congruent disk robots in the plane.

(iii) We show that the problem is FPT parameterized by k, for both target functions, the total number of moves and the total rectilinear traveled length, if we restrict the motion of the centers of the disks to a unit grid. This restriction models problems that arise in real-world applications. For instance, the movement of Amazon Kiva robots [29] is restricted to a floor grid.

2 Preliminaries and Problem Definition

The $\mathcal{O}^*()$ notation hides a polynomial function in the input size n, which is the length of the binary encoding of the instance. We write $[k]$ for the set $\{1,\ldots,k\}$.

A *parameterized problem* Q is a subset of $\Omega^* \times \mathbb{N}$, where Ω is a fixed alphabet. Each instance of Q is a pair (I,κ), where $\kappa \in \mathbb{N}$ is called the *parameter*. A parameterized problem Q is *fixed-parameter tractable* (FPT) [6, 10], if there is an algorithm, called an FPT-*algorithm*, that decides whether an input (I,κ) is a member of Q in time $f(\kappa)\cdot|I|^{\mathcal{O}(1)}$, where f is a computable function and $|I|$ is the input instance size. The class FPT denotes the class of all fixed-parameter tractable parameterized problems.

A parameterized problem Q is FPT-*reducible* to a parameterized problem Q' if there is an algorithm, called an FPT-*reduction*, that transforms each instance (I,κ) of Q into an instance (I',κ') of Q' in time $f(\kappa)\cdot|I|^{\mathcal{O}(1)}$, such that $\kappa' \leqslant g(\kappa)$ and $(I,\kappa) \in Q$ if and only if $(I',\kappa') \in Q'$, where f and g are computable functions. Based on the notion of FPT-reducibility, a hierarchy of parameterized complexity, *the W-hierarchy* $= \bigcup_{t\geqslant 0} \mathsf{W}[t]$, where $\mathsf{W}[t] \subseteq \mathsf{W}[t+1]$ for all $t \geqslant 0$, has been introduced, in which the 0-th level W[0] is the class FPT. The notion of hardness has been defined for each level W[t] of the W-hierarchy for $t \geqslant 1$ [6, 10]. It is commonly believed that W[1] $\neq$ FPT (see [6, 10]).

Let $\mathcal{R} = \{R_i \mid i \in [k]\}$ be a set of robots. We assume, both at the starting and ending positions, that the robots are pairwise non-overlapping (in their interiors). Depending on the problem under consideration, the robots in $\mathcal{R}$ can either be all axis-aligned rectangular robots (not necessarily congruent), or all congruent disk robots of fixed radius, but not a combination of both. We will refer to a robot by its identifying name (e.g., R_i), which determines the location of its center (either the center of the corresponding disk, or the center of the corresponding rectangle – the intersection of its two diagonals) in the schedule at any time step.

For two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$, the *rectilinear* distance, or the *Manhattan* distance, between p and q, is defined as $d_1(p, q) = |x_p - x_q| + |y_p - y_q|$.

A *translation move*, or a *move*, for $R_i \in \mathcal{R}$ w.r.t. a direction $\vec{v}$, is a translation of R_i by a vector $\alpha \cdot \vec{v}$ where $\alpha > 0$; we denote the move as $(R_i, \alpha\vec{v})$. For a vector $\vec{u}$, $\mathsf{translate}(R_i, \vec{u})$ denotes the axis-aligned rectangle resulting from translating R_i by vector $\vec{u}$.

In this paper, we consider only axis-aligned translations, that is translations using the vectors in $\mathcal{V} = \{\vec{H}^-, \vec{H}^+, \vec{V}^-, \vec{V}^+\}$, which are the negative and positive unit vectors of the x- and y-axis, respectively.

We consider two types of moves: *serial* and *coordinated*, where the former type corresponds to the robots moving one at a time (i.e., a robot must finish its move before the next starts), and the latter type corresponds to (possibly) multiple robots moving simultaneously. Formally, a coordinated move is a move in which a subset X of robots move simultaneously and at the same speed. The move ends when all the robots in X reach their desired locations during that move.

We now define collision for the two types of motion.

For a robot R_i that is translated by a vector $\vec{v}$, we say that R_i *collides* with a stationary robot $R_j \neq R_i$, if there exists $0 \leqslant x \leqslant 1$ such that R_j and $\mathsf{translate}(R_i, x \cdot \vec{v})$ intersect in their interior. For two distinct robots R_i and R_j that are simultaneously translated by vectors $\vec{v_i}$ and $\vec{v_j}$, respectively, we say that R_i and R_j *collide* if there exists $0 \leqslant x \leqslant 1$ such that $\mathsf{translate}(R_i, x \cdot \vec{v_i})$ and $\mathsf{translate}(R_j, x \cdot \vec{v_j})$ intersect in their interior.

A *serial schedule* (resp. *coordinated schedule*) $\mathcal{S}$ for $\mathcal{R}$ is a sequence of axis-aligned collision-free serial (resp. coordinated) moves such that after all the moves in $\mathcal{S}$, each R_i ends at its final destination, for $i \in [k]$. The *length* $|\mathcal{S}|$ of the schedule is the number of moves in it, and the cost of $\mathcal{S}$, denoted $cost(\mathcal{S})$, is the total rectilinear distance traveled by all the (centers of the) robots in $\mathcal{S}$. More specifically, suppose that $\mathcal{S} = \langle (R_{i_1}, \vec{v_1}), \ldots, (R_{i_\ell}, \vec{v_\ell}) \rangle$, for some $\ell \in \mathbb{N}$, then $|S| = \ell$, and $cost(\mathcal{S}) = \sum_{j=1}^{\ell} ||\vec{v_j}||$. (Note that each vector in $\mathcal{S}$ either has horizontal or vertical orientation.) In this paper, we study the following problems:

Moves Rectangles Motion Planning (Moves-Rect-MP)
Given: A set of pairwise non-overlapping axis-aligned rectangular robots $\mathcal{R} = \{R_i \mid i \in [k]\}$ each given with its starting and final positions; $k, \lambda \in \mathbb{N}$.
Question: Is there a schedule for $\mathcal{R}$ of length $\leqslant \lambda$?

Let $\mu \geqslant 0$ be a constant.

Grid-Moves μ-Disks Motion Planning (Grid-Moves-μ-Disk-MP)
Given: A set of pairwise non-overlapping congruent disk robots $\mathcal{R} = \{R_i \mid i \in [k]\}$ of radius μ, each given with its starting and final positions; an $N \times M$ unit grid Ω, where $N, M \in \mathbb{N}$; $k, \lambda \in \mathbb{N}$.
Question: Is there a schedule for $\mathcal{R}$ of length at most λ in which the centers of the robots in $\mathcal{R}$ are confined to (moving on) points of Ω?

Cost Rectangles Motion Planning (Cost-Rect-MP)
Given: A set of pairwise non-overlapping axis-aligned rectangular robots $\mathcal{R} = \{R_i \mid i \in [k]\}$ each given with its starting and final positions; $k \in \mathbb{N}$.
Question: Compute a schedule for $\mathcal{R}$ of minimum cost (if one exists).

The Grid-Cost μ-Disks Motion Planning (Grid-Cost-μ-Disk-MP) is defined analogously.

We note that the time complexity for solving the decision problems Moves-Rect-MP and Grid-Moves-μ-Disk-MP will be essentially the same (up to a polynomial factor) as that for solving its optimization version (where we seek to minimize ℓ), as we can binary-search for the length of an optimal schedule.

We also study the Rectangles Coordinated Motion Planning problem (Moves-Rect-CMP) and Disks Coordinated Motion Planning problems (Grid-Moves-μ-Disks-CMP) and their cost counterparts the Cost Rectangles Coordinated Motion Planning (Cost-Rect-CMP) and Cost Disks Coordinated Motion Planning problems (Grid-Cost-μ-Disks-CMP), which are defined analogously with the only difference being that the moves are coordinated. More specifically, the schedule of the robots consists of a sequence of coordinated collision-free moves.

We will use the following Grid-Tiling problem and its varian for our hardness results. Grid-Tiling [6]: Given $k, n \in \mathbb{N}$ and k^2 nonempty sets S_{ij}, $i, j \in [k]$, where each $S_{ij} \subseteq [n] \times [n]$, decide if we can choose a $p_{ij} = (x_{ij}, y_{ij}) \in S_{ij}$, for each $(i, j) \in [k] \times [k]$, such that

- for each $i \in [k]$, $y_{i1} = y_{i2} = \cdots = y_{ik}$, and
- for each $j \in [k]$, $x_{1j} = x_{2j} = \cdots = x_{kj}$.

That is, the y-coordinates of the p_{ij}'s are the same in any row and the x-coordinates are the same in any column. The Increasing Grid-Tiling problem is defined similarly with the exception that we require the sequence of the x-coordinates of $p_{1j}, \ldots, p_{kj}$ and the sequence of y-coordinates of $p_{i1}, \ldots, p_{ik}$ be non-decreasing as opposed to being equal. Both versions of the problem are NP-hard and W[1]-hard parameterized by k [6].

We conclude the preliminaries section by giving upper bounds in k on the length (i.e., number of moves) of an optimal schedule for feasible instances of the problems under consideration. These upper bounds will be used to obtain FPT algorithms for these problems.

Proposition 1 *For any instance of* Cost-Rect-MP*,* Grid-Cost-μ-Disk-MP*,* Cost-Rect-CMP*, and* Grid-Cost-μ-Disks-CMP*, there is an optimal schedule for the instance of length at most* $2k \cdot 5^{k(k-1)}$*.*

3 W[1]-hardness of Moves-Rect-CMP With Obstacles

In this section, we consider the setting where, besides the robots, the plane may contain stationary obstacles. The obstacles are axis-aligned rectangles, and the collision between a robot and an obstacle is defined in the natural way, treating the obstacle as a stationary robot. We show that Moves-Rect-CMP with obstacles is W[1]-hard, contrasting the FPT result for the same problem in the free plane [25].

Theorem 2 Moves-Rect-CMP *with rectangular obstacles is* W[1]*-hard parameterized by the number* k *of robots.*

Proof. We reduce from an instance $(k, n, (S_{ij})_{i,j \in [k]})$ of Grid-Tiling. The produced instance of Moves-Rect-CMP has $2k$ rectangular robots. There are k robots $V_1, \ldots, V_k$ that we refer to as "vertical", which will be responsible for selecting the x-coordinate in each of k columns of the $k \times k$ grid from the Grid-Tiling instance. Each vertical robot has width 1 and height 2. Similarly, there are k "horizontal" robots $H_1, \ldots, H_k$ responsible for selecting the y-coordinate in each of the k rows of the $k \times k$ grid. Each horizontal robot has width 2 and height 1. The goal is now to create an environment, using rectangular obstacles, that encodes the Grid-Tiling instance (see Figure 1). The upper bound

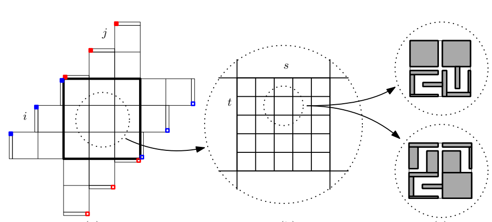

Figure 1: A high-level overview of the constructed instance of Moves-Rect-CMP in Theorem 2 starting from the Grid-Tiling instance with $k = 3$ and $n = 5$.

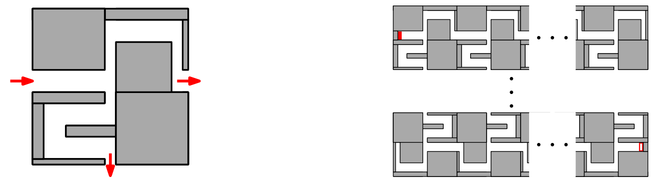

Figure 2: An illustration of a vertical selection gadget.

on the total number of moves is equal to the minimum number of moves each robot needs to reach its destination. The core of the reduction is a $k \times k$ grid (see the bold-boundaried square in the center of Figure 1-(a)). Each of the k^2 cells of this grid is further divided into an $n \times n$ grid, whose cells represent the pairs in the S_{ij}. We assume that the rows are numbered from top to bottom and the columns from left to right; so the t-th row from the top and the s-th column from the left represent the pair (s, t) and depending on whether $(s, t) \in S_{ij}$ or not, the cell contains an "allow-intersection gadget" (Figure 1-(c) bottom) or a "forbid-intersection gadget" (Figure 1-(c) top). To ensure that the horizontal robot H_i and the vertical robot V_j enter the cell at the same time, the starting position of H_i (resp. V_j) is shifted left by $i - 1$ (resp. up by $j - 1$) auxiliary $n \times n$ grids, which are outside of the central $k \times k$ big grid, and hence robots do not cross in these auxiliary grids (see Figure 1-(a)). Each robot then needs to fully pass through $2k - 1$ many $n \times n$ grids. Finally, robots start and end in a selection gadget; see Figure 2 for the vertical selection gadget (the horizontal selection gadget is analogous). The vertical (horizontal) selection gadgets are the only places that allow vertical (horizontal) robots to change their horizontal (vertical) position. That is, a vertical robot with height 2 and width 1 can enter each cell of the top part of the vertical selection gadget from the left and either leave it from the right after 4 moves, or from the bottom after 5 moves. The bottom (top) of the i-th cell in the top (bottom) part of the selection gadget connects directly to an intersection gadget (one of the two possible gadgets in Figure 1-(c)) in the first row and the i-th column of an $n \times n$ grid. Each intersection gadget can be passed by a vertical robot only from top to bottom in 4 moves; a vertical robot does not fit through the left/right entrance of the gadget. Similarly, a horizontal robot can pass through an intersection gadget in 4 moves from left to right, but cannot enter/exit from the top/bottom entrances. The above finishes the description of the gadgets. The total number of moves is then set to $4(n + 1) + 4(2k - 1)n + 3$. Each robot needs $4(2k - 1)n$ moves to pass through $(2k - 1)n$ many intersection gadgets in $2k - 1$ many $n \times n$ grids, and $4(n + 1) + 3$ moves to pass through the selection gadget. For the selection gadget, if the robot leaves from the s-th cell, then it is easy to see from Figure 2 that it makes

$4s+1$ moves in the top part (its $(4s+2)$-nd move counts towards the cell it enters) and $4(n+1-s)+2$ moves in the bottom part of the selection gadget. Hence, robots have to move towards their destinations in every step.

We now show that the original GRID-TILING instance is equivalent to the produced instance of Moves-Rect-CMP with obstacles and with $\lambda = 4(n+1)+4(2k-1)n+3$. We already argued that, in a schedule with an upper bound of λ on the total number of moves, each vertical robot V_j selects a single column in the selection gadgets and then passes through all the intersection gadgets in that column; this selection clearly corresponds to selecting the same x-coordinate in each cell of the j-th column of the $k \times k$ grid (i.e., $x_{1j} = x_{2j} = \cdots = x_{kj}$). Similarly, the horizontal robot H_i picks a single row, effectively ensuring that $y_{i1} = y_{i2} = \cdots = y_{ik}$. To prove the correctness of the reduction, it suffices to show that H_i can pick row t and V_j column s if and only if $(s,t) \in S_{ij}$. For that, let us assume that they pick t and s, respectively, and let us compute the move in which they enter the intersection gadget representing the position (s,t) in the $n \times n$ grid for S_{ij}, where their two paths intersect. First, for V_j to select s, it does $4s+1$ moves in the selection gadget. Afterwards, it does $4(j-1)n$ moves through the auxiliary grids to enter the central $k \times k$ grid. There it does $4 \cdot (i-1) \cdot n$ moves to pass first the $i-1$ many $n \times n$ grids and finally, it does $4 \cdot (t-1)$ moves to reach the top entrance of the corresponding intersection gadget after $4(i+j-2) \cdot n + 4 \cdot (s+t) - 3$ moves. Symmetrically, it is easy to see that H_i is at the left entrance of the intersection gadget after $4(i+j-2) \cdot n + 4 \cdot (s+t) - 3$ many moves as well. It is easy to see that only the allow-intersection gadget (Figure 1-(c) bottom) allows both robots to enter at the same time. This is because vertical robots can pass the crossing point in the same move in which it enters the gadget, while for horizontal robots it is their third move. On the other hand, in the forbid-intersection gadget (Figure 1-(c) top) the crossing is in the first move for both robots and at the same distance from the entrance, and hence there will be a collision. It follows that the instance of Moves-Rect-CMP with obstacles admits a schedule with a total number of moves at most λ if and only if the instance of GRID-TILING is a Yes-instance. ◻

4 On the Cost-Rect-MP Problem

It is not difficult to prove the NP-hardness of Cost-Rect-MP via a reduction from the NP-hard (n^2-1)-puzzle problem [31]. In this section, we present the NP-hardness of Cost-Rect-MP restricted to instances in which each robot must travel along a shortest rectilinear path between its starting and ending positions, denoted Shortest Cost-Rect-MP, which is an important restriction in its own right, and implies the NP-hardness of an important variant of the classical COORDINATED MOTION PLANNING ON GRIDS as well as other problems in this paper:

Theorem 3 SHORTEST COST-RECT-MP *is* NP*-hard.*

We also give FPT algorithms for Cost-Rect-MP and Cost-Rect-CMP, which follow exactly the algorithms given in [24] for Moves-Rect-MP and Moves-Rect-CMP, with a minor tweak of changing the linear program to include an objective function for minimizing the cost, which is a linear function. The running time of the algorithms for Cost-Rect-MP and Cost-Rect-CMP is $\mathcal{O}^*(5^{k^2} \cdot 8^{2k^2 \cdot 5^{k^2}} \cdot 5^{k^3 \cdot 5^{k^2}})$ (see [25]).

5 The FPT Algorithms for Grid-Moves-μ-Disk-MP and Grid-Cost-μ-Disk-MP

The NP-hardness of Grid-Moves-μ-Disk-MP follows from the NP-hardness of the restriction of the problem to instances in which the disks are points (i.e., the radius μ of the disks is zero), proved in [5].

We describe the FPT algorithms for Grid-Moves-μ-Disk-MP and Grid-Cost-μ-Disk-MP. The algorithms are similar to those in the previous section, with the exception that we no longer can use linear programming since the motion must be confined to the grid points (as opposed to the whole plane). Instead, we use Integer Linear Programming (ILP), and that poses quite some complications. To be able to use ILP and obtain fixed-parameter tractability, we resort to Lenstra's result [19,20,26], which requires that the number of variables in the ILP instance be upper bounded by a function of the parameter k. We describe the algorithm for Grid-Moves-μ-Disk-MP; the algorithm for Grid-Cost-μ-Disk-MP is exactly the same except for adding a minimization objective linear function. Since the upper bound on the running time in Lenstra's result [19,20,26] is not practical, we omit discussing the running time of our algorithm and concern ourselves with showing that it is FPT.

We start by guessing the number of moves, λ, in an optimal schedule, which was shown in Section 2 to be upper bounded by a function of k. Next, the algorithm guesses in each step which robot moves and its direction. We create ILP variables to encode the position of each robot in each step, and the amplitude of the translation vector in each move. Since λ and the number of possible directions for each move are upper bounded by a function of k and a constant, respectively, the number of ILP variables is upper bounded by a function of k. What is left is showing that we can add linear constraints to stipulate that the moves are collision-free.

Let R be a disk robot whose center is at (x_0, y_0), and suppose that in a certain move R is translated horizontally by a vector $\overrightarrow{v}$ of amplitude α in the direction $\overrightarrow{H}^+$;

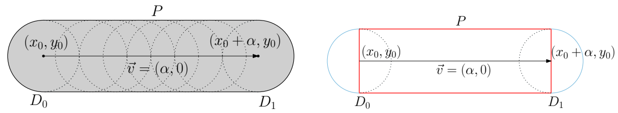

Figure 3: Left: The trace of disk D_0 w.r.t. a translation by a vector $\overrightarrow{v} = (\alpha, 0)$. Right: The decomposition of the trace into the rectangle P (red) and the two disks D_0 and D_1.

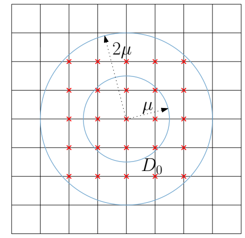

Figure 4: The grid points corresponding to the centers of disks that collide with D_0.

other directions are analogous. At the translation, the center of R is at the point $(x_0 + \alpha, y_0)$. During the motion of R, it covers a region of the plane, denoted *trace*$(R, \overrightarrow{v})$, depicted in Figure 3 (left). For any robot $R' \neq R$, located at point (a, b), R does not collide with R' during this translation iff the disk for R' centered at (a, b) does not intersect the interior of $\mathsf{trace}(R, \overrightarrow{v})$. Note that x_0, y_0, α, a, b are all variables in the ILP instance. To stipulate this non-collision condition, we decompose $\mathsf{trace}(R, \overrightarrow{v})$ into three (non-disjoint) regions (see the right side of Figure 3): the two disks D_0 centered at (x_0, y_0) and D_1 centered at $(x_0 + \alpha, y_0)$ of radius μ, and the interior of rectangle P whose two diametrically-opposite corners/vertices are $(x_0, y_0 + \mu)$ and $(x_0 + \alpha, y_0 - \mu)$. Clearly, R' does not intersect $\mathsf{trace}(R, \overrightarrow{v})$ if and only if it does not intersect any of these three regions. Consider first rectangle P. To stipulate that R' does not intersect the interior of P, we first observe that if R' intersects a vertical line segment of P then it must intersect D_0 or D_1. Since we will later add constraints to stipulate that R' does not intersect D_0 or D_1, it suffices at this point to add constraints to enforce that the interior of R' does not intersect any of the two horizontal line segments of P. We first make a guess for if R' intersects a vertical slab B defined by P such that R' intersects P in B iff it intersects a horizontal segment. If not, it can make intersection with P only around D_0 and D_1, in the other case we do as follows. Observe that if R' does not intersect a horizontal segment of P then (a, b) must be either above the top horizontal segment of P, or below the bottom horizontal segment of P, and in either case, at a distance of at least μ from the corresponding line. We guess which of the two cases holds and add a linear constraint accordingly. For example, if the guess is that (a, b) is above the top horizontal line of P, we add the linear constraint: $b - \mu \geqslant y_0 + \mu$, or equivalently, $b - y_0 \geqslant 2\mu$. The case is similar for the other guess. Since we make two guesses per rectangle R', the number of guesses is FPT for this part. Next, we encode non-collision between R' and D_0; the treatment is analogous for D_1. Since R' and D_0 are congruent disks of radius μ, these disks collide if and only if the distance between their centers is smaller than 2μ. Since μ is a constant and the grid is a unit grid, the number of grid points whose distance from the center (x_0, y_0) of D_0 is smaller than 2μ is a function of μ, this is a constant, and this will be true even after adding some points outside of the vertical slab near the D_i, and those points have coordinates that are linear in x_0 and y_0, and can be enumerated in constant time; see Figure 4. Therefore, we add constraints to enforce that the center (a, b) is not equal to any of these points. To do so, note that two points are not equal if one of the two coordinates of one of the two points is less than the corresponding coordinate of the other point. For any of the candidate points (c, d), we guess which of the four cases holds (i.e., which of the coordinates of (a, b) and (c, d) is less than the corresponding one) and add linear constraints stipulating that. The number of guesses and variables introduced in this part are FPT.

We constructed an ILP instance that can be solved in FPT-time using the results in [19, 20, 26]. Since the number of guesses is FPT, the above nondeterministic algorithm can be simulated in deterministic FPT-time:

Theorem 4 *The* GRID-MOVES-μ-DISK-MP *and the* GRID-COST-μ-DISK-MP *problems are* FPT *parameterized by* k*.*

We can extend the above FPT algorithms to Grid-Moves-μ-Disks-CMP and Grid-Cost-μ-Disks-CMP.

Theorem 5 *The* GRID-MOVES-μ-DISKS-CMP *and the* GRID-COST-μ-DISKS-CMP *problems are* FPT *parameterized by* k*.*

6 Concluding Remarks

A couple of natural open questions ensue from our work:

- Can we extend the $\mathsf{W}[1]$-hardness result of Moves-Rect-CMP where obstacles are present to Moves-Rect-MP (i.e., where the motion is serial)? We note that the coordinated motion was a very essential ingredient in the $\mathsf{W}[1]$-hardness proof for Moves-Rect-CMP.
- Can we extend the FPT algorithms for disk robots confined to a grid, given in this paper, to the case where the disk robots move in the free plane?

References

[1] Manuel Abellanas, Sergey Bereg, Ferran Hurtado, Alfredo García Olaverri, David Rappaport, and Javier Tejel. Moving coins. *Computational Geometry*, 34(1):35–48, 2006.

[2] Pankaj K. Agarwal, Dan Halperin, Micha Sharir, and Alex Steiger. Near-optimal min-sum motion planning for two square robots in a polygonal environment. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4942–4962, 2024.

[3] Jacopo Banfi, Nicola Basilico, and Francesco Amigoni. Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. *IEEE Robotics and Automation Letters*, 2(4):1941–1947, 2017.

[4] Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, David Tolpin, Oded Betzalel, and Solomon Eyal Shimony. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In *IJCAI*, pages 740–746, 2015.

[5] Gruia Călinescu, Adrian Dumitrescu, and János Pach. Reconfigurations in graphs and grids. *SIAM Journal on Discrete Mathematics*, 22(1):124–138, 2008.

[6] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[7] Argyrios Deligkas, Eduard Eiben, Robert Ganian, Iyad Kanj, and M. S. Ramanujan. Parameterized algorithms for coordinated motion planning: Minimizing energy. In *51st International Colloquium on Automata, Languages, and Programming, ICALP*, volume 297 of *LIPIcs*, pages 53:1–53:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[8] Argyrios Deligkas, Eduard Eiben, Robert Ganian, Iyad Kanj, and M. S. Ramanujan. Parameterized algorithms for multiagent pathfinding on trees. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*. ACM, 2025.

[9] Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Henk Meijer, and Christian Scheffer. Coordinated motion planning: Reconfiguring a swarm of labeled robots with bounded stretch. *SIAM Journal on Computing*, 48(6):1727–1762, 2019.

[10] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[11] Adrian Dumitrescu. Mover problems. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 185–211, New York, NY, 2013. Springer New York.

[12] Adrian Dumitrescu and Minghui Jiang. On reconfiguration of disks in the plane and related problems. *Computational Geometry*, 46(3):191–202, 2013.

[13] Eduard Eiben, Robert Ganian, and Iyad Kanj. The parameterized complexity of coordinated motion planning. In *Symposium on Computational Geometry*, volume 258, pages 28:1–28:16, 2023.

[14] Eduard Eiben, Robert Ganian, Iyad Kanj, and M. S. Ramanujan. A minor-testing approach for coordinated motion planning with sliding robots. In *Symposium on Computational Geometry*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[15] Guillermo Esteban, Dan Halperin, Víctor Ruíz, Vera Sacristán, and Rodrigo I. Silveira. Shortest coordinated motion for square robots. In *Proceedings of the 18th International Symposium on Algorithms and Data Structures (WADS)*, volume 14079 of *Lecture Notes in Computer Science*, pages 430–443. Springer, 2023.

[16] Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. Computing coordinated motion plans for robot swarms: The CG: SHOP challenge 2021. *ACM J. Exp. Algorithmics*, 27:3.1:1–3.1:12, 2022.

[17] Foivos Fioravantes, Dusan Knop, Jan Matyás Kristan, Nikolaos Melissinos, and Michal Opler. Exact algorithms and lowerbounds for multiagent path finding: Power of treelike topology. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 17380–17388, 2024.

[18] Foivos Fioravantes, Dusan Knop, Jan Matyás Kristan, Nikolaos Melissinos, and Michal Opler. Exact algorithms for multiagent path finding with communication constraints on tree-like structures. *To appear in AAAI 2025*, abs/2412.08556, 2024. URL: `https://doi.org/10.48550/arXiv.2412.08556`, `arXiv:2412.08556`, `doi:10.48550/ARXIV.2412.08556`.

[19] András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[20] Jr. H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[21] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[22] John E. Hopcroft, Jacob T. Schwartz, and Micha Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "Warehouseman's Problem. *The International Journal of Robotics Research*, 3(4):76–88, 1984.

[23] John E. Hopcroft and Gordon T. Wilfong. Reducing multiple object motion planning to graph searching. *SIAM Journal on Computing*, 15(3):768–785, 1986.

[24] Iyad Kanj and Salman Parsa. On the parameterized complexity of motion planning for rectangular robots. In *40th International Symposium on Computational Geometry, SoCG 2024, June 11-14, 2024, Athens, Greece*, volume 293 of *LIPIcs*, pages 65:1–65:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[25] Iyad Kanj and Salman Parsa. On the parameterized complexity of motion planning for rectangular robots, 2024. URL: https://arxiv.org/abs/2402.17846, arXiv:2402.17846.

[26] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.

[27] David G. Kirkpatrick and Paul Liu. Characterizing minimum-length coordinated motions for two discs. In *Proceedings of the 28th Canadian Conference on Computational Geometry, CCCG*, pages 252–259, 2016.

[28] Daniel Kornhauser, Gary Miller, and Paul Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, pages 241–250, 1984.

[29] Hang Ma and Sven Koenig. AI buzzwords explained: Multi-agent path finding (mapf). 3(3):15–19, 2017.

[30] Geetha Ramanathan and Vangalur S. Alagar. Algorithmic motion planning in robotics: Coordinated motion of several disks amidst polygonal obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 514–522, 1985.

[31] Daniel Ratner and Manfred Warmuth. The $(n^2 - 1)$-puzzle and related relocation problems. *Journal of Symbolic Computation*, 10(2):111–137, 1990.

[32] Jacob T. Schwartz and Micha Sharir. On the piano movers' problem: III. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2:46 – 75, 1983.

[33] Jacob T. Schwartz and Micha Sharir. On the "piano movers'" problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on pure and applied mathematics*, 36(3):345–398, 1983.

[34] Jacob T. Schwartz and Micha Sharir. On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, 4(3):298–351, 1983.

[35] Michael Ian Shamos. *Computational geometry.* PhD thesis, USA, 1978.

[36] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.

[37] Irving Solis, James Motes, Read Sandström, and Nancy M. Amato. Representation-optimal multi-robot motion planning using conflict-based search. *IEEE Robotics Autom. Lett.*, 6(3):4608–4615, 2021.

[38] Kiril Solovey and Dan Halperin. On the hardness of unlabeled multi-robot motion planning. *The International Journal of Robotics Research*, 35(14):1750–1759, 2016.

[39] Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.