

Mover Problems

Adrian Dumitrescu

Abstract Leo Moser asked what is the region of largest area that can be moved around a right-angled corner in a corridor of unit width? Similarly, what is the region of largest area that can be reversed in a T junction made up of roads of unit width? Can a specific 3-dimensional object pass through a given door? Our survey aims at showing that mover problems are no less challenging even if there are no obstacles other than the objects themselves, but there are many objects to move. We survey some recent results on motion planning and reconfiguration for systems of multiple objects and for modular systems with applications in robotics, and collect some open problems coming out of this line of research.

1 Introduction

The motion planning problem in its simplest form is that of finding a collision-free movement for a single object from a given start position to another specified target position in the presence of obstacles; see, e.g., [34]. Besides the pure algorithmic aspect, a natural question is how large such an object can be moved through a given corridor formed by the obstacles. For instance, Leo Moser asked for the region of largest area that can be moved around a right-angled corner in a corridor of unit width [36]; see also [19, G5]. The problem became known as the *piano mover's problem*, or more accurately (since any single mover would be overwhelmed by the task), the *piano movers' problem*.

The motion planning problem can also be formulated for systems of multiple independent objects. Schwartz and Sharir present an algorithm that solves the following motion planning problem that arises in robotics: Given a start and a

Supported in part by NSF CAREER grant CCF-0444188 and NSF grant DMS-1001667.

A. Dumitrescu (✉)

Department of Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI, USA

e-mail: dumitres@uwm.edu.

target configuration of n disks, and a region bounded by a collection of walls, find a continuous motion connecting the two configurations of these bodies during which they avoid collision with the walls and with each other, or establish that no such motion exists [46]. Motion planning for multiple objects has been also addressed in [28]. Again, besides the pure algorithmic aspect, a natural question is to estimate the number of moves (of a certain type) such a reconfiguration requires in terms of n , the number of objects. In this survey we are primarily concerned with this combinatorial aspect of moving multiple objects. We discuss three related reconfiguration scenarios for systems of multiple objects. The reader is also referred to the two surveys on related topics by Pach and Sharir [39, Chap. 9] and by Demaine and Hearn [20].

1. A *body* (or *object*) in the plane is a compact connected set in \mathbb{R}^2 with a nonempty interior. Two initially disjoint bodies *collide* if they share an interior point at some time during their motion. Consider a set of n pairwise interior-disjoint objects in the plane that need to be brought from a given start (initial) configuration S into a desired target (goal) configuration T , without causing collisions. The *reconfiguration* problem for such a system is that of computing a sequence of object motions (a schedule, or motion plan) that achieves this task. Depending on the existence of such a sequence of motions, we call that instance of the problem *feasible* and, respectively, *infeasible*.

Our reconfiguration problem is a simplified version of the multirobot motion planning problem [32], in which a system of robots are operating together in a shared workplace and from time to time need to move from their initial positions to a set of target positions. The workspace is often assumed to extend throughout the entire plane, with no obstacles other than the robots themselves. In many applications, the robots are indistinguishable so any of them can occupy any of the specified target positions. Another application that permits the same abstraction is moving around large sets of heavy objects in a warehouse. Typically, one is interested in minimizing the number of moves and in designing efficient algorithms for carrying out the motion plan. There are several types of moves, such as sliding, translation, or lifting, which lead to three different models that will be discussed in Sect. 2.

2. A different kind of reconfiguration problem appears in connection to *metamorphic* or *self-reconfigurable* modular systems. A *modular* robotic system consists of a number of identical robotic modules that can connect to, disconnect from, and relocate relative to adjacent modules; see examples in [14, 18, 37, 38, 40, 45, 48–50, 53]. Typically, the modules have a regular symmetry so that they can be packed densely, with small gaps between them. Such a system can be viewed as a large swarm of physically connected robotic modules that behave collectively as a single entity.

The system changes its overall shape and functionality by reconfiguring into different formations. In most cases individual modules are not capable of moving by themselves; however, the entire system may be able to move to a new position when its members repeatedly change their positions relative to

their neighbors, by rotating or sliding around other modules [13, 37, 52], or by expansion and contraction [45]. In this way the entire system, by changing its aggregate geometric structure, may acquire new functionalities to accomplish a given task or to interact with the environment.

Shape changing in these composite systems is envisioned as a means to accomplish various tasks, such as reconnaissance, exploration, satellite recovery, or operation in constrained environments inaccessible to humans (e.g., nuclear reactors, space, or deep water). For another example, a self-reconfigurable robot can aggregate as a snake to traverse a tunnel and then reconfigure as a six-legged spider to move over uneven terrain. A novel useful application is to realize self-repair: A self-reconfigurable robot carrying some additional modules may abandon the failed modules and replace them with spare units [45]. It is usually assumed that the modules must remain connected all (or most) of the time during reconfiguration.

The motion planning problem for such a system is that of computing a sequence of module motions that brings the system in a given initial configuration I into a desired goal configuration G . Reconfiguration for modular systems acting in a grid-like environment, and where moves must maintain connectivity of the whole system has been studied in [25–27], focusing on two basic capabilities of such systems: reconfiguration and locomotion. We present details in Sect. 3.

3. In some cases the problem of bringing a set of pairwise disjoint objects (in the plane or in the space) to a desired goal, configuration, admits the following abstraction: We have an underlying finite or infinite connected graph; the start configuration is represented by a set of n chips at n start vertices and the target configuration by another set of n target vertices. A vertex can be both a start and a target position. The case when the chips are labeled or unlabeled gives two different variants of the problem. In one move a chip can follow an arbitrary path in the graph and end up at another vertex, provided the path (including the end vertex) is free of other chips [16].

The motion planning problem for such a system is that computing a sequence of chip motions that brings the chips from their initial positions to their target positions. Again, the problem may be feasible or infeasible. We address multiple aspects of this variant in Sect. 4. We note that the three models mentioned earlier in (1) do not fall in the above graph reconfiguration framework, because an object may partially overlap several target positions.

2 Models of Reconfiguration for Systems of Objects in the Plane

We formulate these models for systems of disks, since they are simpler and most of our results are for this class of convex bodies. These rules can be extended (not necessarily uniquely) for arbitrary convex bodies in the plane. The decision

problems we refer to below, pertaining to various reconfiguration problems we discuss here, are in standard form and concern systems of (arbitrary or congruent) disks. For instance, the *reconfiguration problem* U-SLIDE-RP for congruent disks is as follows: Given a start configuration and a target configuration, each with n unlabeled congruent disks in the plane, and a positive integer k , is there a reconfiguration motion plan with at most k sliding moves? It is worth clarifying that for the unlabeled variant, if the start and target configuration contain subsets of congruent disks, there is freedom in choosing which disks will occupy target positions. However, in the labeled variant, this assignment is uniquely determined by the labeling; of course, a valid labeling must respect the size of the disks.

1. *Sliding model*: One move is sliding a disk to another location in the plane without colliding with any other disk, where the disk center moves along an arbitrary continuous curve. This model was introduced in [8]. The labeled and unlabeled variants are L-SLIDE-RP and U-SLIDE-RP, respectively.
2. *Translation model*: One move is translating a disk to another location in the plane along a fixed direction without colliding with any other disk. This is a restriction imposed to the sliding model above for making each move as simple as possible. This model was introduced in [2]. The labeled and unlabeled variants are L-TRANS-RP and U-TRANS-RP, respectively.
3. *Lifting model*: One move is lifting a disk and placing it back in the plane anywhere in the free space, that is, at a location where it does not intersect (the interior of) any other disk. This model was introduced in [7]. The labeled and unlabeled variants are L-LIFT-RP and U-LIFT-RP, respectively.

It turns out that moving a set of objects from one place to another is related to certain *separability* problems [12, 17, 29, 31]; see also [44]. For instance, given a set of disjoint polygons in the plane, may each be moved “to infinity” in a continuous motion in the plane without colliding with the others? Often constraints are imposed on the types of motions allowed, e.g., only translations, or only translations in a fixed set of directions. Usually only one object is permitted to move at a time. Without the convexity assumption on the objects, it is easy to show examples when the objects are interlocked and could only be moved “together” in the plane; however, they could be easily separated using the third dimension, i.e., in the lifting model.

It can be shown that for the class of disks, the reconfiguration problem in each of these models is always feasible [2, 7, 8, 12, 29, 31]. This follows essentially from the feasibility in the sliding model and the translation model; see Sect. 2.1. For the more general class of convex objects, one needs to allow rotations. For simplicity, we restrict ourselves mostly to the case of disks. We are thus led to the following generic question: Given a pair of start and target configurations, each consisting of n pairwise disjoint disks in the plane, what is the minimum number of moves that suffice for transforming the start configuration into the target configuration for each of these models?

If no target disk coincides with a start disk, so each disk must move at least once, obviously at least n moves are required. In general, one can use (a variant of) the following simple *universal* algorithm for the reconfiguration of n objects

Fig. 1 $2n - 1$ moves are necessary (in the sliding or the lifting model) to bring the n segments from vertical position to horizontal position

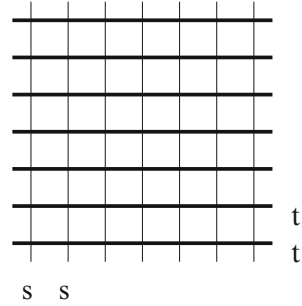


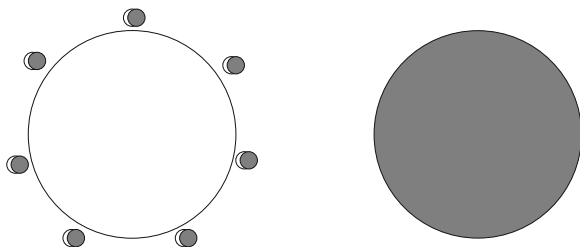
Table 1 Comparison summary: number of moves for disks in the plane/chips in the grid

Model	Type	Lower bound	Upper bound
Translating disks	Congruent (unlabeled)	$\lfloor 5n/3 \rfloor - 1$	$2n - 1$
	Arbitrary (unlabeled)	$2n$	$2n$
Sliding disks	Congruent (unlabeled)	$16n/15 - o(n)$	$3n/2 + o(n)$
	Arbitrary (unlabeled)	$2n - o(n)$	$2n - 1$
Lifting disks	Congruent (unlabeled)	$n + \Omega(n^{1/2})$	$n + O(n^{2/3})$
	Arbitrary (unlabeled)	$\lfloor 5n/3 \rfloor$	$9n/5$
Sliding chips (grid)	Unlabeled	n	n
	Labeled	$3n/2$	$7n/4$
Lifting chips (grid)	Unlabeled	n	n
	Labeled	$3n/2$	$3n/2$

using $2n$ moves. To be specific, consider the lifting model. In the first step (n moves), move all the objects away anywhere in the free space. In the second step (n moves), bring the objects “back” to target positions. For the class of segments (or rectangles) as objects, it is easy to construct examples that require $2n - 1$ moves for reconfiguration, in any of the three models, even for congruent objects; see Fig. 1. A first goal is to estimate more precisely where in the interval $[n, 2n]$ the answer lies for each of these models. The best current lower and upper bounds on the number of moves necessary in the three models described earlier are listed in Table 1. It is quite interesting to compare the bounds on the number of moves for the three models, translation, sliding, and lifting, with those for the graph variants discussed in Sect. 4. Table 1, which is constructed on the basis of the results in [2, 7, 8, 16], facilitates this comparison.

Some remarks are in order. Clearly, any lower bound (on the number of moves) for lifting is also valid for sliding, and any upper bound (on the number of moves) for sliding is also valid for lifting. Another observation is that for lifting, those objects whose target position coincides with their start position can be safely ignored, while for sliding this is not true. A simple example is illustrated in Fig. 2: Assume that we have a large disk surrounded by $n - 1$ smaller ones. The large disk has to be moved to another location, while the $n - 1$ smaller disks have to stay where they are.

Fig. 2 One move is enough in the lifting model, while $n - 1$ are needed in the sliding model for this pair of start and target configurations with n disks each (here $n = 8$). Start disks are *white* and target disks are *shaded*



One move is enough in the lifting model, while $n - 1$ are needed in the sliding model: One needs to make space for the large disk to move out by moving out about half of the small disks and then moving them back in to the same positions.

A move is a *target move* if it moves a disk to a final target position. Otherwise, it is a *nontarget* move. Our lower bounds use the following argument: If no target disk coincides with a start disk (so each disk must move), a schedule that requires x nontarget moves must consist of at least $n + x$ moves.

2.1 The Sliding Model

It is not difficult to show that, for the class of disks, the reconfiguration problem in the sliding model is always feasible. More generally, the problem remains feasible for the class of all convex objects using sliding moves; this follows from Theorem 1. This old result appears in the work of Fejes Tóth and Heppes [29], but it can be traced back to de Bruijn [12]; some algorithmic aspects of the problem have been addressed subsequently by Guibas and Yao [31].

Theorem 1 ([12,29,31]). *Any set of n convex objects in the plane can be separated via translations all parallel to any given fixed direction, with each object moving once only. If the topmost and bottommost points of each object are given (or can be computed in $O(n \log n)$ time), an ordering of the moves can be computed in $O(n \log n)$ time.*

The universal algorithm mentioned earlier can be adapted to perform the reconfiguration of any set of n convex objects. It performs $2n$ moves for the reconfiguration of n disks. In the first step (n moves), in decreasing order of the x -coordinates of their centers, slide the disks initially along a horizontal direction, one by one to the far right. Observe that no collisions can occur. In the second step (n moves), bring the disks “back” to target positions in increasing order of the x -coordinates of their centers. (General convex objects may need rotations and translations in the second step.) Already for the class of disks, Theorem 3 shows that one cannot do much better in terms of the number of moves. The following bounds on the number of moves for translating disks are due to Bereg et al. [8] (Theorems 2 and 3).

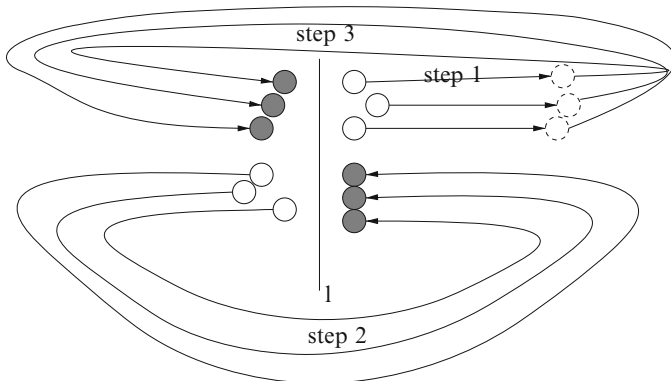


Fig. 3 Algorithm with three steps for sliding congruent disks. The start disks are *white* and the target disks are *shaded*

Theorem 2 ([8]). *Given a pair of start and target configurations S and T , each consisting of n congruent disks, $\frac{3n}{2} + O(\sqrt{n \log n})$ sliding moves always suffice for transforming the start configuration into the target configuration. The entire motion can be computed in $O(n^{3/2}(\log n)^{-1/2})$ time. On the other hand, there exist pairs of configurations that require $(1 + \frac{1}{15})n - O(\sqrt{n})$ moves for this task.*

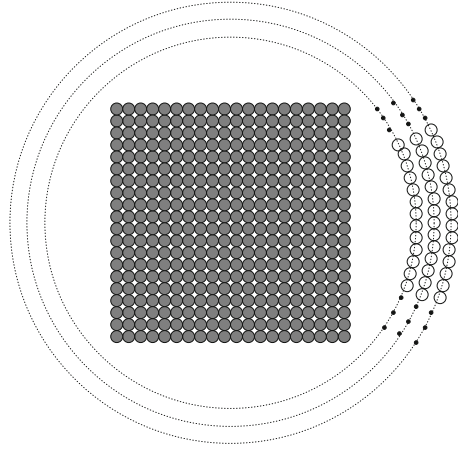
We now briefly sketch the upper bound proof and the corresponding algorithm in [8] for congruent disks. First, one shows the existence of a line bisecting the set of centers of the start disks such that the strip of width 6 around this line contains a small number of disks. More precisely, the following holds:

Lemma 2.1 ([8]). *Let S be a set of n pairwise disjoint unit (radius) disks in the plane. Then there exists a line ℓ that bisects the centers of the disks such that the parallel strip of width 6 around ℓ (that is, ℓ runs in the middle of this strip) contains entirely at most $O(\sqrt{n \log n})$ disks.*

Let S' and T' be the centers of the start disks and target disks, respectively, and let ℓ be the line guaranteed by Lemma 2.1. We can assume that ℓ is vertical. Denote by $s_1 = \lfloor n/2 \rfloor$ and $s_2 = \lceil n/2 \rceil$ the number of centers of start disks to the left and right of ℓ (centers on ℓ can be assigned to the left or right). Let $m = O(\sqrt{n \log n})$ be the number of start disks contained entirely in the vertical strip of width 6 around ℓ . Denote by t_1 and t_2 the number of centers of target disks to the left and right of ℓ , respectively. By symmetry, we can assume that $t_1 \leq n/2 \leq t_2$.

Let R be a region containing all start and target disks, e.g., the smallest axis-aligned rectangle that contains all disks. The algorithm has three steps. All moves in the region R are taken along horizontal lines, i.e., perpendicularly to the line ℓ . The reconfiguration procedure is schematically shown in Fig. 3. This illustration ignores the disks/targets in the center parallel strip.

Fig. 4 A lower bound of $n + \Omega(\sqrt{n})$ moves: $\Theta(\sqrt{n})$ rings, each with $\Theta(\sqrt{n})$ start disk positions. Targets are densely packed in a square formation enclosed by the rings



- Step 1* Slide to the far right all start disks whose centers are to the right of ℓ and the (other) start disks in the strip, one by one, in decreasing order of their x -coordinates (with ties broken arbitrarily). At this point all $t_2 \geq n/2$ target disk positions whose centers lie right of ℓ are free.
- Step 2* In increasing order of their x -coordinates, fill free target positions whose centers are right of ℓ using all the $s'_1 \leq s_1 \leq n/2$ remaining disks whose centers are to the left of ℓ . These disks are taken in increasing order of their x -coordinates: Each disk translates first to the left, then to the right on a wide arc, and to the left again in the end. Note that $s'_1 \leq n/2 \leq t_2$. Now all the target positions whose centers lie left of ℓ are free.
- Step 3* Move to place the far away disks: First continue to fill target positions whose centers are to the right of ℓ , in increasing order of their x -coordinates. When done, fill target positions whose centers are left of ℓ , in decreasing order of their x -coordinates. Note that at this point all target positions whose centers lie left of ℓ are free.

The only nontarget moves are those done in step 1 and their number is $n/2 + O(\sqrt{n \log n})$, so the total number of moves is $3n/2 + O(\sqrt{n \log n})$.

A first idea in constructing a lower bound is as follows: The target configuration consists of a set of n densely packed unit (radius) disks contained, for example, in a square of side length $\approx 2\sqrt{n}$. The disks in the start configuration enclose the target positions using concentric rings, that is, $\Theta(\sqrt{n})$ rings, each with $\Theta(\sqrt{n})$ start disk positions, as shown in Fig. 4. Now observe that for each ring, the first move that involves a disk in that ring must be a nontarget move. The number of rings is $\Theta(\sqrt{n})$, from which a lower bound of $n + \Omega(\sqrt{n})$ follows.

This basic idea of a cage-like construction can be further refined by redesigning the cage [8]. The new design is more complicated and uses “rigidity” considerations, which go back to stable disk packings of density 0 due to Böröczky [9]. A packing \mathcal{C} of unit (radius) disks in the plane is said to be *stable* if each disk is kept fixed

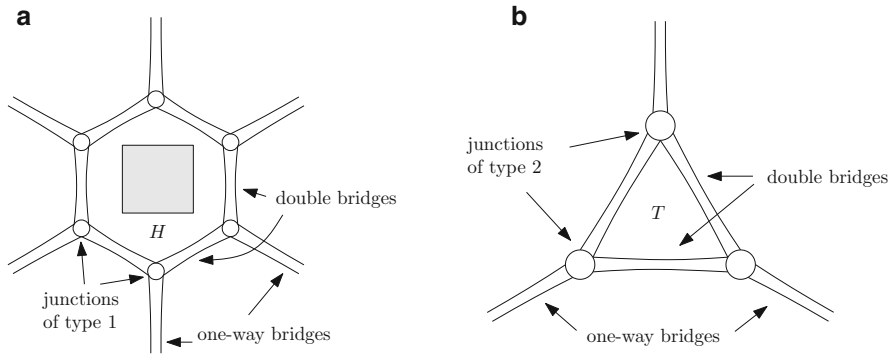


Fig. 5 Two start configurations based on hexagonal and triangular cage-like constructions. Targets are densely packed in a square formation enclosed by the cage, as shown in (a); those in (b) are labeled “T”

by its neighbors [11]. More precisely, \mathcal{C} is stable if none of its elements can be translated by any small distance in any direction without colliding with the others. It is easy to see that any stable system of (unit) disks in the plane must have infinitely many elements. Böröczky [9] showed that somewhat surprisingly, there exist stable systems of unit disks with an arbitrarily small density (i.e., the total area of the disks inside any sufficiently large axis-aligned square is arbitrarily small). These systems can be adapted for the purpose of constructing a lower bound in the sliding model for congruent disks. The details are quite technical, and we only sketch here the new cage-like constructions depicted in Fig. 5.

Let’s refer to the disks in the start (respectively target) configuration as white (respectively, black) disks. Now fix a large n , and take n white disks. Use $O(\sqrt{n})$ of them to build three junctions connected by three “double-bridges” to enclose a triangular region that can accommodate n tightly packed nonoverlapping black disks, as shown in Fig. 5b. Divide the remaining white disks into three roughly equal groups, each of size $\frac{n}{3} - O(\sqrt{n})$, and rearrange each group to form the initial section of a “one-way bridge” attached to the unused sides of the junctions. Each of these bridges consists of five rows of disks of “length” roughly $\frac{n}{15}$, where the length of a bridge is the number of disks along its side. The design of both the junctions and the bridges prohibits any target move before one moves out a sequence of about $\frac{1}{5} \cdot \frac{n}{3} = \frac{n}{15}$ white adjacent disks starting at the far end of one of the one-way bridges. The reason is that with the exception of the at most $3 \times 4 = 12$ white disks at the far ends of the truncated one-way bridges, every white disk is fixed by its neighbors. The total number of necessary moves is at least $(1 + \frac{1}{15})n - O(\sqrt{n})$ for this triangular ring construction, and at least $(1 + \frac{1}{30})n - O(\sqrt{n})$ for the hexagonal ring construction. Observe that the triangular cage yields a better bound.

For disks of arbitrary radii, the following result is obtained by the same authors [8]:

Theorem 3 ([8]). *Given a pair of start and target configurations, each consisting of n disks of arbitrary radii, $2n$ sliding moves always suffice for transforming the*

Fig. 6 A simple configuration that requires about $3n/2$ moves (basic step for the recursive construction)

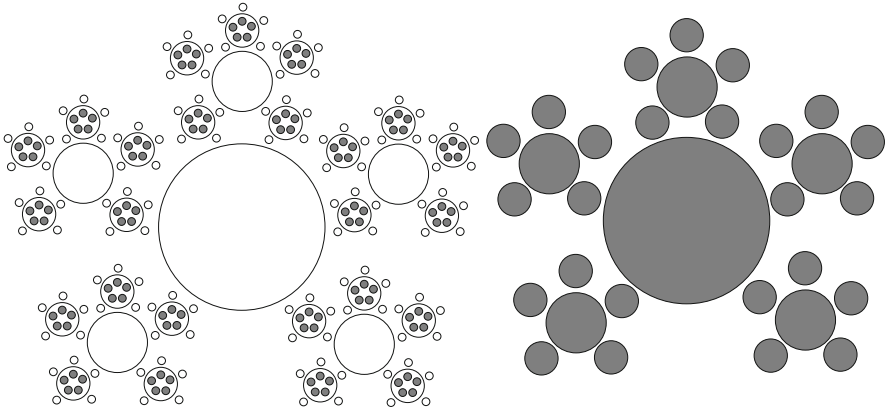
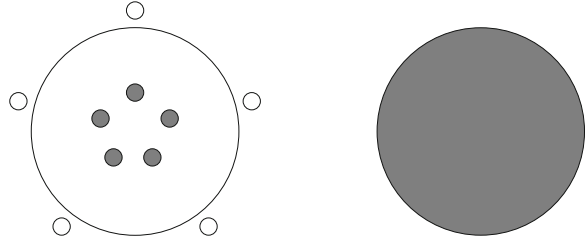


Fig. 7 Recursive lower-bound construction for sliding disks of arbitrary radii: $m = 2$ and $k = 3$

start configuration into the target configuration. The entire motion can be computed in $O(n \log n)$ time. On the other hand, there exist pairs of configurations that require $2n - o(n)$ moves for this task.

The upper bound follows from the universal reconfiguration algorithm described earlier. The lower bound is a recursive construction shown in Fig. 7. It is obtained by iterating recursively the basic construction in Fig. 6, which requires about $3n/2$ moves: Note that the target positions of the $n - 1$ small disks lie inside the start position of the large disk. This means that no small disk can reach its target before the large disk moves away, that is, before roughly half of the $n - 1$ small disks move away. So about $n/2$ nontarget moves are necessary; thus, about $3n/2$ moves in total are necessary.

In the recursive construction, the small disks around a large one are replaced by the “same” construction scaled (see Fig. 7). All disks have distinct radii, so it may be convenient to think of them as being labeled. There are one large disk and $2m + 1$ groups of smaller disks around it close to the vertices of a regular $(2m + 1)$ -gon ($m \geq 1$). The small disks on the last level or recursion have targets inside the big ones they surround (the other disks have targets somewhere else). Let $m \geq 1$ be fixed. If there are k levels in the recursion, then instead of about $n/2$ nontarget moves

Fig. 8 A two-disk configuration that requires 4 translation moves

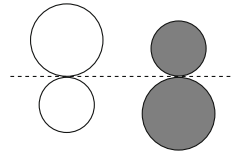
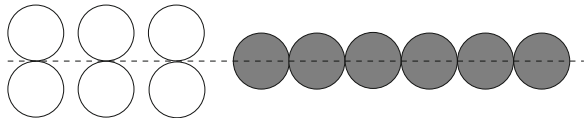


Fig. 9 A configuration of 6 congruent disks that requires 9 translation moves



being necessary (as previously argued for Fig. 6), now about $n/2 + n/4 + \dots + n/2^k$ nontarget moves are necessary. The precise calculation for $m = 1$ yields the lower bound $2n - O(n^{\log_3 2}) = 2n - O(n^{0.631})$; see [8].

2.2 The Translation Model

This model, first studied by Abellanas et al. [2], is a constrained variant of the sliding model in which each move is a translation along a fixed direction; that is, the center of the moving disk traces a line segment. With some care, one can modify the universal algorithm mentioned in the introduction and find a suitable order in which disks can be moved “to infinity” and then moved “back” to target position via translations almost parallel to any given fixed direction using $2n$ translation moves [2].

That this bound is best possible for arbitrary radii disks can be easily seen in Fig. 8. Consider a pair of start and target configurations with two disks each. The two start disks and the two target positions are tangent to the same line. Note that the first move cannot be a target move. Assume that the larger disk moves first, and observe that its location must be above the horizontal line. If the second move is again a nontarget move, we have accounted for 4 moves already. Otherwise, no matter which disk moves to its target position, the other disk will require two more moves to reach its target. The situation when the smaller disk moves first is analogous. One can repeat this basic configuration with two disks, using different radii, to obtain configurations with an arbitrary large (even) number of disks, which require $2n$ translation moves.

Theorem 4 ([2]). *Given a pair of start and target configurations, each consisting of n disks of arbitrary radii, $2n$ translation moves always suffice for transforming the start configuration into the target configuration. On the other hand, there exist pairs of configurations that require $2n$ such moves.*

For congruent disks, the configuration shown in Fig. 9 (the first lower bound that was proposed) requires $3n/2$ moves, since from each pair of tangent disks, the first move must be a nontarget move [42]. A better lower bound, $\lfloor 8n/5 \rfloor$, due

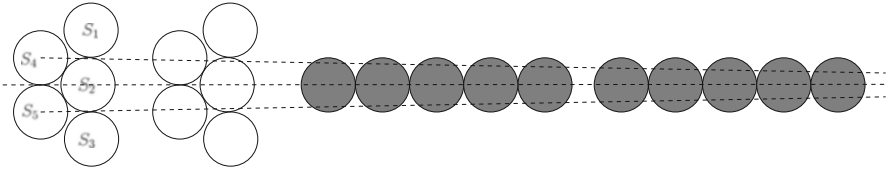


Fig. 10 Reconfiguration of a system of 10 congruent disks that needs 16 translation moves. Start disks are *white* and target disks are *shaded*

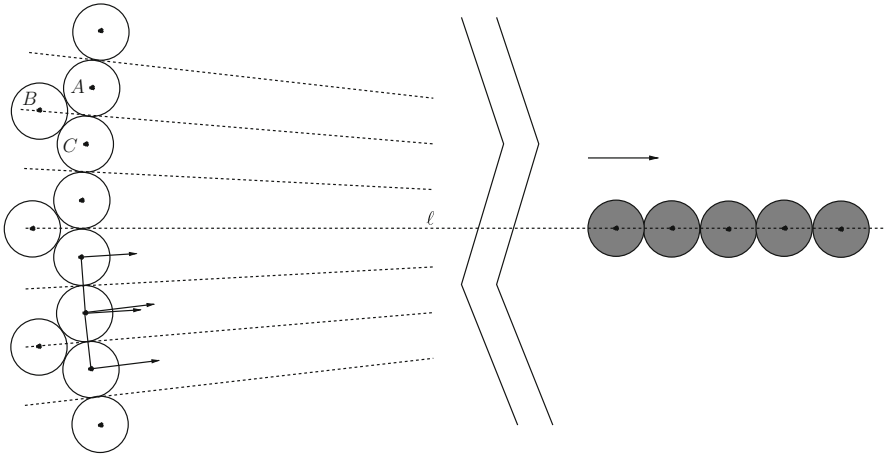
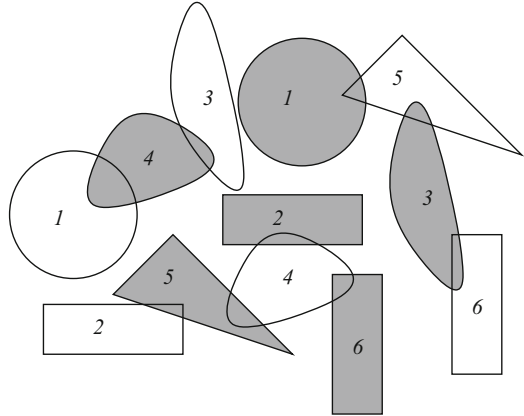


Fig. 11 Illustration of the lower-bound construction for translating congruent unlabeled disks, for $m = 3, n = 11$. The disks are *white* and their targets are *shaded*. Two consecutive partially overlapping parallel strips of width 2 are shown

to Abellanas et al. [2] is illustrated in Fig. 10. The construction is symmetric with respect to the middle horizontal line. Here we have groups of five disks each, where to “move” one group to some five target positions requires eight translation moves. In each group, the disks $S_2, S_4,$ and S_5 are pairwise tangent, and S_1 and S_3 are each tangent to S_2 ; the tangency lines in the latter pairs are almost horizontal, converging to the middle horizontal line. Here are two different ways for moving one group, each requiring three nontarget moves: (a) S_1 and S_3 move out, S_2 moves to destination, S_4 moves out, S_5 moves to destinations followed by the rest. (b) $S_4, S_5,$ and S_2 move out (to the left), S_1 and S_3 move to destinations followed by the rest.

The current best lower bound, $\lfloor 5n/3 \rfloor - 1$, is due to Dumitrescu and Jiang [24]. Let $n = 3m + 2$. The start and target configurations, each with n disks, are shown in Fig. 11. The n target positions are all on a horizontal line ℓ , with the disks at these positions forming a horizontal chain, T_1, \dots, T_n , consecutive disks being tangent to each other. Let o denote the center of the median disk, $T_{\lfloor n/2 \rfloor}$. Let $r > 0$ be very large. The start disks are placed on two very slightly convex chains (two concentric circular arcs):

Fig. 12 Reconfiguration of convex bodies with translations. The start positions are *unshaded*; the target positions are *shaded*



- $2m + 2$ disks in the first layer (chain). Their centers are $2m + 2$ equidistant points on a circular arc of radius r centered at o .
- m disks in the second layer. Their centers are m equidistant points on a concentric circular arc of radius $r \cos \alpha + \sqrt{3}$. Each pair of consecutive points on the circle of radius r subtends an angle of 2α from the center of the circle (α is very small).

The parameters of the construction are chosen to satisfy $\sin \alpha = 1/r$ and $2n \sin n\alpha \leq 2$. Set, for instance, $\alpha = 1/n^2$, which results in $r = \Theta(n^2)$. Alternatively, the configuration can be viewed as consisting of m groups of three disks each, plus two disks, one at the higher and one at the lower end of the chain along the circle of radius r . We summarize the bounds for translation of congruent disks in the following theorem.

Theorem 5 ([2, 24]). *Given a pair of start and target configurations, each consisting of n congruent disks, $2n - 1$ translation moves always suffice for transforming the start configuration into the target configuration. On the other hand, there exist pairs of configurations that require $\lfloor 5n/3 \rfloor - 1$ such moves.*

Translating Convex Bodies. We briefly discuss the general problem of reconfiguration of convex bodies with translations. Refer to Fig. 12. When the convex bodies have different shapes, sizes, and orientations, we assume that the correspondence between the start positions $\{S_1, \dots, S_n\}$ and the target positions $\{T_1, \dots, T_n\}$ is given explicitly: T_i is a translated copy of S_i . In other words, we deal with the labeled variant of the problem. Theorem 6 can be easily extended to the unlabeled variant by first computing a valid correspondence by shape matching. The $2n$ upper bound for translating arbitrary disks can be extended to arbitrary convex bodies in the plane.

Theorem 6 ([24]). *For the reconfiguration with translations of n labeled disjoint convex bodies in the plane, $2n$ moves are always sufficient and sometimes necessary.*

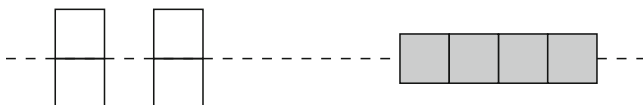


Fig. 13 A lower bound of $\lfloor 3n/2 \rfloor$ for translating axis-parallel unit squares. The start positions (grouped in pairs) are tangent to the x -axis, which intersects the target positions (*shaded*). Each of the target squares is symmetric about the x -axis

Translating Unlabeled Axis-Parallel Unit Squares. Throughout a translation move, the moving square remains axis-parallel; however, the move can be in any direction. The construction in Fig. 13 gives a lower bound of $\lfloor 3n/2 \rfloor$ and we have the following bounds.

Theorem 7 ([24]). *For the reconfiguration with translations of n unlabeled axis-parallel unit squares in the plane, $2n - 1$ moves are always sufficient, and $\lfloor 3n/2 \rfloor$ moves are sometimes necessary.*

Recently García et al. estimated the number of moves necessary for the reconfiguration of n axis-parallel rectangles, where each move is a collision-free axis-parallel translation [30].

2.3 The Lifting Model

For systems of n congruent disks, one can estimate the number of moves that are always sufficient with higher accuracy. The following bounds were established by Dumitrescu and Bereg [7]:

Theorem 8 ([7]). *Given a pair of start and target configurations S and T , each with n congruent disks, one can move the disks from S to T using $n + O(n^{2/3})$ moves in the lifting model. The entire motion can be computed in $O(n \log n)$ time. On the other hand, for each n , there exist pairs of configurations that require $n + \Omega(n^{1/2})$ moves for this task.*

The lower-bound construction is illustrated in Fig. 14 for $n = 25$. Assume for simplicity that $n = m^2$, where m is odd. We place the disks of T onto a grid $X \times X$ of size $m \times m$, where $X = \{2, 4, \dots, 2m\}$. We place the disks of S onto a grid of size $(m - 1) \times (m - 1)$ so that they overlap with the disks from T . The grid of target disks contains $4m - 4$ disks on its boundary. We “block” them with $2m - 2$ start disks in S by placing them so that each start disk overlaps with two boundary target disks as shown in the figure. We place the last start disk somewhere else, and we have accounted for $(m - 1)^2 + (2m - 2) + 1 = m^2$ start disks. As proved in [7], at least $n + \lfloor m/2 \rfloor$ moves are necessary for reconfiguration (it can be verified that this number of lifting moves suffices for this construction).

The proof of the upper bound of $n + O(n^{2/3})$ is technically more complicated. It builds a binary space partition of the plane into convex polygonal (bounded or

Fig. 14 A pair of start and target configurations, each with $n = 25$ congruent disks, which require 27 lifting moves. The start disks are *white* and the target disks are *shaded*

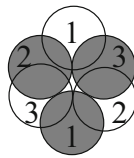
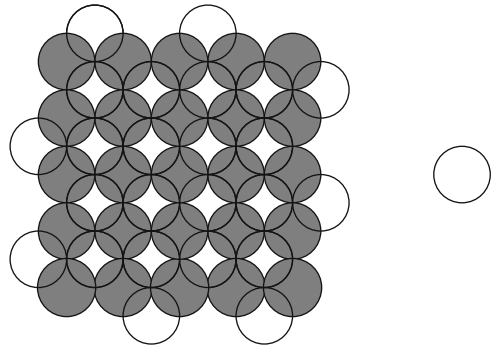


Fig. 15 A group of three disks (of distinct radii) that require five moves to reach their targets; part of the lower-bound construction for lifting disks of arbitrary radii. The disks are *white* and their targets are *shaded*

unbounded) regions satisfying certain properties. Once the partition is obtained, a *shifting* algorithm moves disks from some regions to fill the target positions in other regions; see [7] for details. Since the disks whose target position coincides with their start position can be safely ignored in the beginning, the upper bound yields an efficient algorithm that performs a number of moves close to the optimum (for large n).

For arbitrary radius disks, the following result is obtained in [7].

Theorem 9 ([7]). *Given a pair of start and target configurations S and T , each with n disks with arbitrary radii, $9n/5$ moves always suffice for transforming the start configuration into the target configuration. On the other hand, for each n , there exist pairs of configurations that require $\lfloor 5n/3 \rfloor$ moves for this task.*

The lower bound is very simple. We use disks of different radii (although the radii can be chosen very close to the same value if desired). Since all disks have distinct radii, one can think of the disks as being labeled. Consider the set of three disks, labeled 1, 2, and 3 in Fig. 15. The two start and target disks labeled i are congruent, for $i = 1, 2, 3$. To transform the start configuration into the target configuration takes at least two nontarget moves, thus five moves in total. By repeatedly using groups of three (with different radii), one gets a lower bound of $5n/3$ moves, when n is a multiple of three, and $\lfloor 5n/3 \rfloor$ in general.

We now explain the approach in [7] for the upper bound for n disks of arbitrary radii. Let $S = \{s_1, \dots, s_n\}$ and $T = \{t_1, \dots, t_n\}$ be the start and target configurations.

We assume that for each i , disk s_i is congruent to disk t_i ; i.e., t_i is the target position of s_i . If the correspondence $s_i \rightarrow t_i$ is not given (but only the two sets of disks), it can be easily computed by sorting both S and T by radius.

In a directed graph $D = (V, E)$, let $d_v = d_v^+ + d_v^-$ denote the degree of vertex v , where d_v^+ is the out-degree of v and d_v^- is the in-degree of v . Let $\beta(D)$ be the maximum size of a subset V' of V such that $D[V']$, the subgraph induced by V' , is acyclic. In [7] the following inequality is proved for any directed graph:

$$\beta(D) \geq \max \left(\sum_{v \in V} \frac{1}{d_v^+ + 1}, \sum_{v \in V} \frac{1}{d_v^- + 1} \right).$$

For a disk ω , let $\overset{\circ}{\omega}$ denote the interior of ω . Let S be a set of k pairwise disjoint red disks, and T be a set of l pairwise disjoint blue disks. Consider the bipartite red–blue disk intersection graph $G = (S, T, E)$, where $E = \{(s, t) : s \in S, t \in T, \overset{\circ}{s} \cap \overset{\circ}{t} \neq \emptyset\}$. Using the triangle inequality (among sides and diagonals in a convex quadrilateral), one can easily show that any red–blue disk intersection graph $G = (S, T, E)$ is planar, and consequently $|E| \leq 2(|S| + |T|) - 4 = 4n - 4$. We think of the start and target disks being labeled from 1 to n , so that the target of start disk i is target disk i . Consider the directed *blocking graph* $D = (S, F)$ on the set S of n start disks, where

$$F = \{(s_i, s_j) : i \neq j \text{ and } \overset{\circ}{s}_i \cap \overset{\circ}{s}_j \neq \emptyset\}.$$

If $(s_i, s_j) \in F$, we say that disk i *blocks* disk j . (Note that $s_i \cap t_i \neq \emptyset$ does not generate any edge in D .) Since if $(s_i, s_j) \in F$, then $(s_i, t_j) \in E$, we have $|F| \leq |E| \leq 4n - 4$. The algorithm first eliminates all the directed cycles from D using some nontarget moves, and then fills the remaining targets using only target moves. Let

$$d^+ = \frac{\sum_{v \in S} d_v^+}{n} = \frac{|F|}{n}$$

be the average out-degree in D . We have $d^+ \leq 4$, which further implies (by Jensen’s inequality)

$$\beta(D) \geq \sum_{v \in S} \frac{1}{d_v^+ + 1} \geq \frac{n}{d^+ + 1} \geq \frac{n}{5}.$$

Let $S' \subset S$ be a set of disks of size at least $n/5$ and whose induced subgraph is acyclic in D . Move out far away the remaining set S'' of at most $4n/5$ disks, and note that the faraway disks do not block any of the disks in S' . Perform a topological sort on the acyclic graph $D[S']$ induced by S' , and fill the targets of these disks in that order using only target moves. Then fill the targets with the faraway disks in any order. The number of moves is at most $n + 4n/5 = 9n/5$, as claimed. Figure 16 shows the bipartite intersection graph G and the directed blocking graph D for a small example, with the corresponding reconfiguration procedure explained above.

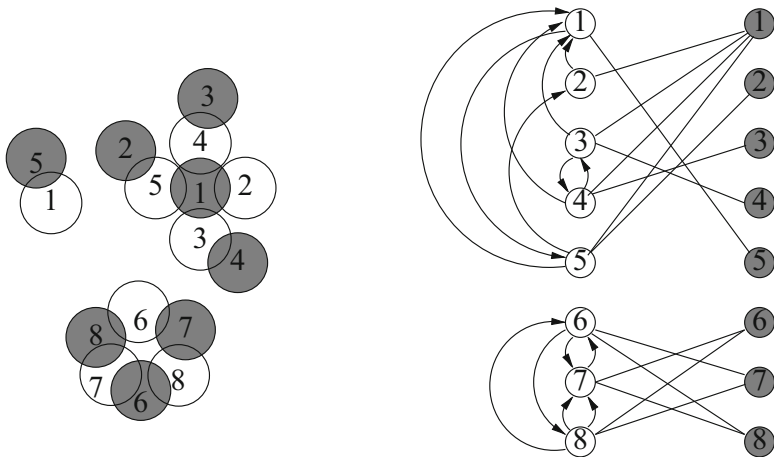


Fig. 16 The bipartite intersection graph G and the directed blocking graph D . Move out 4, 5, 7, 8; no cycles remain in D . Fill targets 3, 2, 1, 6, and then 4, 5, 7, 8. The start disks are *white* and the target disks are *shaded*

Similar to the case of congruent disks, the resulting algorithm performs a number of moves that is not more than a constant times the optimum. Indeed, as mentioned in the introduction, the disks whose target positions coincide with their start positions can be safely ignored. So each of the remaining start disk needs to be moved from its initial place. Without loss of generality, all disks need to move, so the number of moves in any solution is at least n (the case when some disks are ignored is only better). Since the algorithm above performs at most $9n/5$ lifting moves, it achieves a constant approximation ratio, $9/5$.

Computational Complexity of Optimal Reconfiguration. While, as discussed, reconfiguration with $2n$ or fewer moves is always possible in any of the three models (sliding, translation, and lifting), optimal reconfiguration (employing a minimum number of moves) is probably NP-hard in each of these models. This has been confirmed for translation and sliding by Dumitrescu and Jiang [24]: (a) Both the labeled and unlabeled versions of the disk reconfiguration problem with translations U-TRANS-RP and L-TRANS-RP are NP-hard even for congruent disks. (b) Both the labeled and unlabeled versions of the sliding disks reconfiguration problem in the plane U-SLIDE-RP and L-SLIDE-RP are NP-hard even for congruent disks.

2.4 Further Questions

We list a few open problems concerning the three models discussed:

1. Reduce the gap between the $16n/15 - o(n)$ lower bound and the $3n/2 + o(n)$ upper bound on the number of moves for sliding n congruent disks.

2. Consider the reconfiguration problem for congruent squares (with arbitrary orientation) in the sliding model. It can be checked that the $3n/2 + o(n)$ upper bound for congruent disks still holds in this case; however, the $16n/15 - o(n)$ lower bound based on stable disk packings cannot be used. Observe that the $n + \Omega(n^{1/2})$ lower bound for congruent disks in the lifting model (Fig. 14) can be realized with congruent (even axis-aligned) squares and therefore holds for congruent squares in the sliding model as well. Can one deduce better bounds for this variant?
3. Derive a $(2 - \delta)n$ upper bound for translating congruent disks (where δ is a positive constant), or improve the (multiplicative constant in the) $\lfloor 5n/3 \rfloor - 1$ lower bound.
4. Consider the reconfiguration problem for congruent *labeled* disks in the sliding model. It is easy to see that the $\lfloor 5n/3 \rfloor$ lower bound holds, since the construction in Fig. 15 can be realized with congruent disks. Find a $(2 - \delta)n$ upper bound (where δ is a positive constant), or improve the (multiplicative constant in the) $\lfloor 5n/3 \rfloor$ lower bound.
5. The type of construction in Fig. 13 has been used previously for disks to obtain the first lower bound of $\lfloor 3n/2 \rfloor$ for translating unit disks [42]. It is interesting to note that neither of the two subsequent improved constructions, $\lfloor 8n/5 \rfloor$ of Abellanas et al. [2], or ours $\lfloor 5n/3 \rfloor - 1$ in Theorem 5, seems to work for squares.
6. Reduce the gap between the $\lfloor 5n/3 \rfloor$ lower bound and the $9n/5$ upper bound on the number of moves for lifting n disks of arbitrary radii.
7. What is the computational complexity of the reconfiguration problem in the lifting model? Are U-LIFT-RP and L-LIFT-RP NP-hard for unit disks?

3 Modular and Reconfigurable Systems

A number of issues related to motion planning and analysis of rectangular metamorphic robotic systems are addressed in [26]. A distributed algorithm for reconfiguration that applies to a relatively large subclass of configurations, called horizontally convex configurations, is presented. Several fundamental questions in the analysis of metamorphic systems have been also addressed. In particular, the following two questions have been shown to be decidable: (a) whether a given set of motion rules maintains connectivity; (b) whether a goal configuration is reachable from a given initial configuration (at specified locations).

For illustration, we present the *rectangular model* of metamorphic systems introduced in [25–27]. Consider a plane that is partitioned into an integer grid of square cells indexed by their center coordinates in the underlying x - y coordinate system. This partition of the plane is only a useful abstraction, as the module-size determines the grid size in practice. At any time, each cell may be empty or occupied by a module. The *reconfiguration* of a metamorphic system consisting of n modules is a sequence of configurations of the modules in the grid at discrete time steps $t = 0, 1, 2, \dots$; see below. Let V_t be the configuration of the modules at time t , where

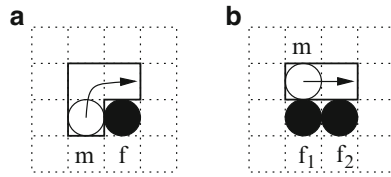


Fig. 17 Moves in the rectangular model: (a) clockwise *NE* rotation and (b) sliding in the *E* direction. Fixed modules are *shaded*. The cells in which the moves take place are outlined in the figure

we often identify V_t with the set of cells occupied by the modules or with the set of their centers. A useful feature we insist on is maintaining connectivity; i.e., for each t , the graph $G_t = (V_t, E_t)$ must be connected, where for any t , E_t is the set of edges connecting pairs of cells in V_t that are side-adjacent. The following two generic motion rules (Fig. 17) define the rectangular model [25–27]. These rules can be applied in all axis-parallel orientations, in fact generating 16 rules, eight for rotation and eight for sliding. A somewhat similar model is presented in [13].

- *Rotation*: A module m side-adjacent to a stationary module f rotates through an angle of 90° around f either clockwise or counterclockwise. Figure 17a shows a clockwise *NE* rotation. For rotation to take place, both the target cell and the cell at the corresponding corner of f that m passes through (*NW* in the given example) have to be empty.
- *Sliding*: Let f_1 and f_2 be stationary cells that are side-adjacent. A module m that is side-adjacent to f_1 and adjacent to f_2 slides along the sides of f_1 and f_2 into the cell that is adjacent to f_1 and side-adjacent to f_2 . Figure 17b shows a sliding move in the *E* direction. For sliding to take place, the target cell has to be empty.

The system may execute moves sequentially, when only one module moves at each discrete time step, or concurrently (when more modules can move at each discrete time step). Concurrent execution has the advantage to speed up the reconfiguration process. If concurrent moves are allowed, additional conditions have to be imposed, as in [26, 27]. In order to ensure motion precision, each move is guided by one or two modules that are stationary during the same step. The following result of Dumitrescu and Pach settles a conjecture formulated in [26].

Theorem 10 ([25]). *The set of motion rules of the rectangular model guarantees the feasibility of motion planning for any pair of connected configurations V and V' having the same number of modules. That is, following the above rules, V and V' can always be transformed into each other so that all intermediate configurations are connected.*

The algorithm in [25] is far from being intuitive or straightforward. The main difficulties that have to be overcome are dealing with holes and avoiding certain deadlock situations during reconfiguration. The proof of correctness of the algorithm and the analysis of the number of moves (cubic in the number of modules,

Fig. 18 A rotation move that temporarily disconnects the configuration

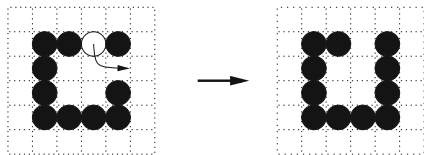
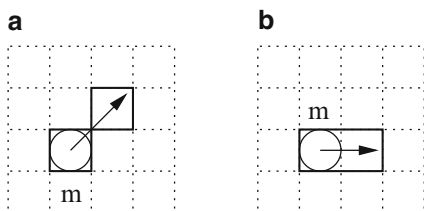


Fig. 19 Moves in the weak rectangular model: (a) *NE* diagonal move and (b) side move in the *E* direction. The cells in which the moves take place are outlined in the figure



for sequential execution) are also quite involved. It is easy to construct examples so that neither sliding nor rotation alone can reconfigure them to straight chains; here a *straight chain* is a set of modules that form a straight line chain in the grid. However, conform with Theorem 10, the motion rules of the rectangular model (rotation and sliding, Fig. 17) are sufficient to guarantee reachability while maintaining the system connected at each discrete time step. This had been proved earlier only for the special class of horizontally convex systems [26].

A somewhat different model can be obtained if, instead of the connectedness requirement at each time step, one imposes the *single backbone* condition [26]: Each move of a module (sliding or rotation) is along the single connected backbone formed by the other modules. If concurrent moves are allowed, additional conditions have to be imposed, as in [26]. A subtle difference exists between requiring the configuration to be connected at each discrete time step and requiring the existence of a connected backbone along which a module slides or rotates. A one step motion that does not satisfy the single backbone condition appears in Fig. 18: The initial connected configuration temporarily disconnects during the move and reconnects at the end of it. The algorithm from [25] has the nice property that the single backbone condition is satisfied during the whole procedure.

It is worth briefly discussing another rectangular model for which the result in Theorem 10 holds. The following two generic motion rules (Fig. 19) define the *weak rectangular model*. These rules are again applicable in all axis-parallel orientations, in fact generating eight rules, four diagonal moves and four side (axis-parallel) moves. The only requirement is that the configurations must remain connected at each discrete time step.

- *Diagonal move*: A module m moves diagonally to an empty cell corner adjacent to $cell(m)$.
- *Side move*: A module m moves to an empty cell side adjacent to $cell(m)$.

The same result from Theorem 10 holds for this second model [25]; however, its proof and corresponding reconfiguration algorithm are much simpler.

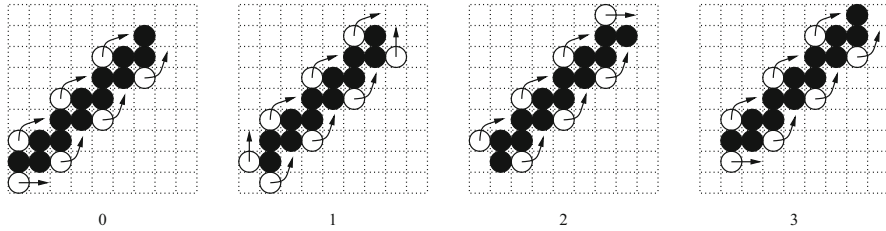


Fig. 20 Formation of 20 modules moving diagonally at a speed of $\frac{1}{3}$ (diagonal formation)

Theorem 11 ([25]). *The set of motion rules of the weak rectangular model guarantees the feasibility of motion planning for any pair of connected configurations having the same number of modules.*

It is worth mentioning that reconfigurations in the rectangular model can be also viewed in the broader context of transformations of binary images. Bose et al. recently studied several such transformations while insisting on maintaining the connectivity of both the foreground and the background of the images in each step [10].

A different variant of interrobot reconfiguration is useful in applications for which there is no clear preference between the use of a single large robot versus a group of smaller ones [15]. This leads to the merging of individual smaller robots into a larger one or the splitting of a large robot into smaller ones. For example, in a surveillance or rescue mission, a large robot is required to travel to a designated location in a short time. Then the robot may create a group of small robots that are to explore concurrently a large area. Once the task is complete, the robots might merge back into the large robot that carried them.

As mentioned in [22], there is considerable research interest in the task of having one autonomous vehicle follow another, and in general in studying robots moving in formation. Dumitrescu et al. [27] examined the problem of dynamic self-reconfiguration of a modular robotic system to a formation aimed at reaching a specified target position as quickly as possible. A number of fast formations for both rectangular and hexagonal systems are presented, achieving a constant-ratio guarantee on the time to reach a given target in the asymptotic sense. For example, a snake-like formation (with $n \geq 4$ modules, n even) can move at a speed of $\frac{1}{3}$ in the rectangular model. In Fig. 20 the formation at time 0 reappears at time 3 translated diagonally by one unit. Thus, by repeatedly going through these configurations, the modules can move in the *NE* direction at a speed of $\frac{1}{3}$.

We conclude this section with some remaining questions on modular and reconfigurable systems related to the results presented. Preliminary results of Abel and Kominers [1] suggest that the first two questions below have a positive answer. This, however, remains to be confirmed.

1. The reconfiguration algorithm in the rectangular model makes at most $2n^3$ moves in the worst case [26]. On the other hand, the reconfiguration of a vertical chain

into a horizontal chain requires only $\Theta(n^2)$ moves, and it is believed that no other pair of configurations requires more. A quadratic upper bound on the number of moves has been proved in the weak rectangular model [25], but it remains open in the first model.

2. Study whether the analogous rules of rotation and sliding in three dimensions permit the feasibility of motion planning for any pair of connected configurations having the same number of modules.
3. If concurrent execution is permitted (under appropriate assumptions), what is the smallest number of concurrent steps that suffices for the reconfiguration of any pair of connected configurations V and V' with n modules—in the rectangular model and in the weak variant? Is it $O(n)$? (This has been shown for a special class of horizontally convex systems [26] and, e.g., in a related 3D rectangular model studied in [4].)

4 Reconfigurations in Graphs and Grids

In certain applications, objects are indistinguishable, and the chips are therefore unlabeled; for instance, a modular robotic system consists of a number of identical modules (robots), each having identical capabilities [25–27]. In other applications the chips may be labeled. The variant with unlabeled chips is easier and always feasible, while the variant with labeled chips may be infeasible: A classical example is the 15-puzzle on a 4×4 grid—introduced by Sam Loyd in 1878—which admits a solution if and only if the start permutation is an even permutation [33, 47]. Most of the work done so far concerns labeled versions of the reconfiguration problem, and here we give only a short account.

For the generalization of the 15-puzzle on an arbitrary graph (with $k = v - 1$ labeled chips in a connected graph on v vertices), Wilson [51] gave an efficiently checkable characterization of the solvable instances of the problem. Kornhauser et al. have extended his result to any $k \leq v - 1$ and provided bounds on the number of moves for solving any solvable instance [35]. Ratner and Warmuth showed that finding a solution with a minimum number of moves for the $(N \times N)$ -extension of the 15-puzzle is NP-hard [43], so the reconfiguration problem in graphs with labeled chips is NP-hard. Auletta et al. gave a linear-time algorithm for the *pebble motion on a tree* problem [5]. This problem is the labeled variant of the same reconfiguration problem studied in [16]; however, each move is along one edge only. Papadimitriou et al. studied a problem of *motion planning on a graph* in which there is a mobile robot at one of the vertices, say s , that has to reach a designated vertex t using the smallest number of moves, in the presence of obstacles (pebbles) at some of the other vertices [41]. Robot and obstacle moves are done along edges, and obstacles have no destination assigned and may end up in any vertex of the graph. The problem has been shown to be NP-complete even for planar graphs, and a polynomial-time approximation algorithm with ratio $O(\sqrt{n})$ was given in [41].

The following results are proved in [16] for the “chips in graph” reconfiguration problem described in part (3) of Sect. 1. Recall that in one move a chip can follow a path in the graph and end up at another vertex, provided the path (including the end vertex) is free of other chips.

- (a) The reconfiguration problem in graphs with unlabeled chips U-GRAPH-RP is NP-hard, and even APX-hard.
- (b) The reconfiguration problem in graphs with labeled chips L-GRAPH-RP is APX-hard.
- (c) For the infinite planar rectangular grid, both the labeled and unlabeled variants L-GRID-RP and U-GRID-RP are NP-hard.
- (d) There is a ratio-3 approximation algorithm for the unlabeled version in graphs U-GRAPH-RP. Thereby one gets a ratio-3 approximation algorithm for the unlabeled version U-GRID-RP in the (infinite) rectangular grid.
- (e) It can be shown that n moves are always enough (and sometimes necessary) for the reconfiguration of n unlabeled chips in graphs. For the case of trees, a linear-time algorithm that performs an optimal (minimum) number of moves is presented.
- (f) It is shown that $7n/4$ moves are always enough, and $3n/2$ are sometimes necessary, for the reconfiguration of n labeled chips in the infinite planar rectangular grid (L-GRID-RP).

Next, we give some details showing that in the infinite grid, n moves always suffice for the reconfiguration of n unlabeled chips, and of course it is easy to construct tight examples. The result holds in a more general graph setting [item (5) in the above list]: Let G be a connected graph, and let V and V' two n -element subsets of its vertex set $V(G)$. Imagine that we place a chip at each element of V and we want to move them into the positions of V' (V and V' may have common elements). A move is defined as shifting a chip from v_1 to v_2 [$v_1, v_2 \in V(G)$] along a path in G so that no intermediate vertices are occupied.

Theorem 12 ([16]). *In G one can get from any n -element initial configuration V to any n -element final configuration V' using at most n moves, so that no chip moves twice.*

It suffices to prove the theorem for trees, and we'd like to include the short proof here. We argue by induction on the number of chips. Take the smallest tree T containing V and V' , and consider an arbitrary leaf l of T . Assume first that the leaf l belongs to V : say $l = v$. If v also belongs to V' , the result trivially follows by induction, so assume that this is not the case. Choose a path P in T , connecting v to an element v' of V' such that no internal point of P belongs to V' . Apply the induction hypothesis to $V \setminus \{v\}$ and $V' \setminus \{v'\}$ to obtain a sequence of at most $n - 1$ moves, and add a final (unobstructed) move from v to v' .

The remaining case when the leaf l belongs to V' is symmetric: Say $l = v'$; choose a path P in T , connecting v' to an element v of V such that no internal

point of P belongs to V . First move v to v' and append the sequence of at most $n - 1$ moves obtained from the induction hypothesis applied to $V \setminus \{v\}$ and $V' \setminus \{v'\}$. This completes the proof.

Theorem 12 implies that in the infinite rectangular grid, we can get from any starting position to any ending position of the same size n in at most n moves. It is interesting to compare this to the problem of sliding congruent unlabeled disks in the plane, where one can come up with cage-like constructions that require about $\frac{16n}{15}$ moves [8], as discussed in Sect. 2.1. We conclude this section with two remaining questions on reconfigurations in graphs and grids:

1. Can the ratio-3 approximation algorithm for the unlabeled version in graphs U-GRAPH-RP be improved? Is there an approximation algorithm with a better ratio for the infinite planar rectangular grid?
2. Is it possible to close or reduce the gap between the $3n/2$ lower bound and the $7n/4$ upper bound on the number of moves for the reconfiguration of n labeled chips in the infinite planar rectangular grid?

5 Conclusion

The different reconfiguration models discussed in this survey have raised new interesting mathematical questions and revealed surprising connections with other older ones. For instance, the key ideas in the reconfiguration algorithm in [25] were derived from the properties of a system of *maximal cycles*, similar to those of the block decomposition of graphs [21]. The lower-bound configuration with unit disks for the sliding model in [8] uses “rigidity” considerations and properties of *stable* packings of disks studied a long time ago by Böröczky [9]; in particular, he showed the existence of stable systems of unit disks with arbitrarily small density. A suitable modification of his construction yields our lower bound.

The study of the lifting model offered other interesting connections: The algorithm for unit disks given in [7] is intimately related to the notion of a *center point* of a finite point set and to the following property derived from it: Given two sets each with n pairwise disjoint unit disks, there exists a binary space partition of the plane into polygonal regions each containing roughly the same small number ($\approx n^{2/3}$) of disks and such that the total number of disks intersecting the boundaries of the regions is small ($\approx n^{2/3}$). The reconfiguration algorithm for disks of arbitrary radius relies on a new lower bound on the maximum order of induced *acyclic* subgraphs of a directed graph [7], analogous to the bound on the *independence number* of an undirected graph given by Turán’s theorem [3]. Moreover, we have used the crucial fact that a bipartite disk intersection graph (drawn as a geometric graph on the set of disk centers) is planar, to obtain a linear upper bound on its number of edges. Finally, the ratio-3 approximation algorithm for the unlabeled version in graphs is obtained by applying the *local ratio method* of Bar-Yehuda [6] to a graph H constructed from the given graph G .

Regarding the various models of reconfiguration for systems of objects in the plane, we have presented estimates on the number of moves that are necessary in the worst case. From the practical viewpoint, one would like to convert these estimates into approximation algorithms with a good ratio guarantee. As shown for the lifting model, the upper-bound estimates on the number of moves give good approximation algorithms for large values of n . However, further work is needed in this direction for the sliding model and the translation model in particular.

Note. An earlier survey [23] on this topic came out with many misprints and errors that were introduced by the publisher. This prompted the author to prepare the current updated version.

Acknowledgements The author is grateful to his collaborators, S. Bereg, G. Călinescu, M. Jiang, J. Pach, I. Suzuki, and M. Yamashita, for allowing him to borrow some material from their respective joint works.

References

1. Z. Abel, S.D. Kominers, Pushing hypercubes around. Preprint (2008). [arXiv:0802.3414v2](https://arxiv.org/abs/0802.3414v2)
2. M. Abellanas, S. Bereg, F. Hurtado, A.G. Olaverri, D. Rappaport, J. Tejel, Moving coins. *Comput. Geom.: Theor. Appl.* **34**(1), 35–48 (2006)
3. N. Alon, J. Spencer, *The Probabilistic Method*, 2nd edn. (Wiley, New York, 2000)
4. G. Aloupis, S. Collette, M. Damian, E.D. Demaine, R.Y. Flatland, S. Langerman, J. O'Rourke, S. Ramaswami, V. Sacristán Adinolfi, S. Wuhler, Linear reconfiguration of cube-style modular robots. *Comput. Geom.: Theor. Appl.* **42**(6–7), 652–663 (2009)
5. V. Auletta, A. Monti, M. Parente, P. Persiano, A linear-time algorithm for the feasibility of pebble motion in trees. *Algorithmica* **23**, 223–245 (1999)
6. R. Bar-Yehuda, One for the price of two: a unified approach for approximating covering problems. *Algorithmica* **27**, 131–144 (2000)
7. S. Bereg, A. Dumitrescu, The lifting model for reconfiguration. *Discr. Comput. Geom.* **35**(4), 653–669 (2006)
8. S. Bereg, A. Dumitrescu, J. Pach, Sliding disks in the plane. *Int. J. Comput. Geom. Appl.* **15**(8), 373–387 (2008)
9. K. Böröczky, Über stabile Kreis- und Kugelsysteme. *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.* **7**, 79–82 (1964) [in German]
10. P. Bose, V. Dujmović, F. Hurtado, P. Morin, Connectivity-preserving transformations of binary images. *Comput. Vis. Image Understand.* **113**, 1027–1038 (2009)
11. P. Braß, W. Moser, J. Pach, *Research Problems in Discrete Geometry* (Springer, New York, 2005)
12. N.G. de Bruijn, Aufgaben 17 and 18. *Nieuw Archief voor Wiskunde* **2**, 67 (1954) (in Dutch)
13. Z. Butler, K. Kotay, D. Rus, K. Tomita, Generic decentralized control for a class of self-reconfigurable robots, in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington, DC, May 2002, pp. 809–816
14. A. Casal, M. Yim, Self-reconfiguration planning for a class of modular robots, in *Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing (SPIE'99)*, 1999, pp. 246–256
15. A. Castano, P. Will, A polymorphic robot team, in *Robot Teams: From diversity to Polymorphism*, ed. by T. Balch, L.E. Parker (A K Peters, 2002), pp. 139–160

16. G. Călinescu, A. Dumitrescu, J. Pach, Reconfigurations in graphs and grids. *SIAM J. Discr. Math.* **22**(1), 124–138 (2008)
17. B. Chazelle, T.A. Ottmann, E. Soisalon-Soininen, D. Wood, The complexity and decidability of separation, in *Proceedings of the 11th International Colloquium on Automata, Languages and Programming (ICALP '84)*. LNCS, vol. 172 (Springer-Verlag, New York, 1984), pp. 119–127
18. G. Chirikjian, A. Pamecha, I. Ebert-Uphoff, Evaluating efficiency of self-reconfiguration in a class of modular robots. *J. Robot. Syst.* **13**(5), 317–338 (1996)
19. H.T. Croft, K.J. Falconer, R.K. Guy, *Unsolved Problems in Geometry* (Springer-Verlag, New York, 1991)
20. E.D. Demaine, R.A. Hearn, Games, puzzles and computation. A K Peters: I-IX, 1–237 (2009)
21. R. Diestel, *Graph Theory, 2nd edn., Graduate Texts in Mathematics*, vol. 173 (Springer-Verlag, New York, 2000)
22. G. Dudek, M. Jenkin, E. Milios, A taxonomy of multirobot systems, in *Robot Teams: From diversity to Polymorphism*, ed. by T. Balch, L.E. Parker (A K Peters, 2002), pp. 3–22
23. A. Dumitrescu, Motion planning and reconfiguration for systems of multiple objects, in *Mobile Robots: Perception & Navigation*, ed. by S. Kolski. Advanced Robotic Systems, Kirchengasse 43/3, A-1070 Vienna, Austria, EU, 523–542 (2007)
24. A. Dumitrescu, M. Jiang, On reconfiguration of disks in the plane and other related problems, in *Proceedings of the 23rd International Symposium on Algorithms and Data Structures (WADS'09)*, Banff, Alberta, Canada, August 2009. LNCS, vol. 5664 (Springer, New York, 2009), pp. 254–265
25. A. Dumitrescu, J. Pach, Pushing squares around. *Graphs Comb.* **22**(1), 37–50 (2006)
26. A. Dumitrescu, I. Suzuki, M. Yamashita, Motion planning for metamorphic systems: feasibility, decidability and distributed reconfiguration. *IEEE Trans. Robot. Autom.* **20**(3), 409–418 (2004)
27. A. Dumitrescu, I. Suzuki, M. Yamashita, Formations for fast locomotion of metamorphic robotic systems. *Int. J. Robot. Res.* **23**(6), 583–593 (2004)
28. M. Erdmann, T. Lozano-Pérez, On multiple moving objects. *Algorithmica* **2**(1–4), 477–521 (1987)
29. L. Fejes Tóth, A. Heppes, Über stabile Körpersysteme. *Compos. Math.* **15**, 119–126 (1963) [in German]
30. A. García, C. Huemer, F. Hurtado, J. Tejel, Moving rectangles, in *Proceedings of the VII Jornadas de Matemática Discreta y Algorítmica*, Santander, 2010
31. L. Guibas, F.F. Yao, On translating a set of rectangles, in *Computational Geometry*, ed. by F. Preparata. Advances in Computing Research, vol. 1 (JAI Press, London, 1983), pp. 61–67
32. Y. Hwuang, N. Ahuja, Gross motion planning—a survey. *ACM Comput. Surv.* **25**(3), 219–291 (1992)
33. W.W. Johnson, Notes on the 15 puzzle. *I. Am. J. Math.* **2**, 397–399 (1879)
34. K. Kedem, M. Sharir, An efficient motion-planning algorithm for a convex polygonal object. *Discr. Comput. Geom.* **5**, 43–75 (1990)
35. D. Kornhauser, G. Miller, P. Spirakis, Coordinating pebble motion on graphs, the diameter of permutation groups, and applications, in *Proceedings of the 25th Symposium on Foundations of Computer Science (FOCS'84)*, Singer Island, FL, pp. 241–250 (1984)
36. L. Moser, Problem 66–11. *SIAM Rev.* **8**, 381 (1966)
37. S. Murata, H. Kurokawa, S. Kokaji, Self-assembling machine, in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'94)*, San Diego, CA, USA, pp. 441–448 (1994)
38. A. Nguyen, L.J. Guibas, M. Yim, Controlled module density helps reconfiguration planning, in *Proceedings of IEEE International Workshop on Algorithmic Foundations of Robotics (WAFR'00)*, Dartmouth College Hanover, NH (2000)
39. J. Pach, M. Sharir, *Combinatorial Geometry and Its Algorithmic Applications—The Alcalá Lectures* (American Mathematical Society, Providence, RI, 2009)
40. A. Pamecha, I. Ebert-Uphoff, G. Chirikjian, Useful metrics for modular robot motion planning. *IEEE Trans. Robot. Autom.* **13**(4), 531–545 (1997)

41. C. Papadimitriou, P. Raghavan, M. Sudan, H. Tamaki, Motion planning on a graph, in *Proceedings of the 35th Symposium on Foundations of Computer Science (FOCS'94)*, pp. 511–520
42. D. Rappaport, *Communication at the 16th Canadian Conference on Computational Geometry (CCCG'04)*, August 2004
43. D. Ratner, M. Warmuth, Finding a shortest solution for the $(N \times N)$ -extension of the 15-puzzle is intractable. *J. Symb. Comput.* **10**, 111–137 (1990)
44. J. O'Rourke, *Computational Geometry in C*, 2nd edn. (Cambridge University Press, Cambridge, 1998)
45. D. Rus, M. Vona, Crystalline robots: self-reconfiguration with compressible unit modules. *Autonomous Robots* **10**, 107–124 (2001)
46. J.T. Schwartz, M. Sharir, On the “piano mover’s” problem, I. *Commun. Pure Appl. Math.* **36**, 345–398 (1983); II. *Adv. Appl. Math.* **4**, 298–351 (1983); III. *Int. J. Robot. Res.* **2**, 46–75 (1983); V. *Commun. Pure Appl. Math.* **37**, 815–848 (1984)
47. W.E. Story, Notes on the 15 puzzle. II. *Am. J. Math.* **2**, 399–404 (1879)
48. J.E. Walter, J.L. Welch, N.M. Amato, Distributed reconfiguration of metamorphic robot chains, in *Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC'00)*, July 2000, Portland, Oregon, pp. 171–180
49. J.E. Walter, J.L. Welch, N.M. Amato, Reconfiguration of hexagonal metamorphic robots in two dimensions, in *Sensor Fusion and Decentralized Control in Robotic Systems III*, ed. by G.T. McKee, P.S. Schenker. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, vol. 4196, 2000, pp. 441–453
50. X. Wei, W. Shi-gang, J. Huai-wei, Z. Zhi-zhou, H. Lin, Strategies and methods of constructing self-organizing metamorphic robots, in *Mobile Robots: New Research*, ed. by J.X. Liu (Nova Science, 2005), pp. 1–38
51. R.M. Wilson, Graph puzzles, homotopy, and the alternating group. *J. Comb. Theor. Ser. B* **16**, 86–96 (1974)
52. M. Yim, Y. Zhang, J. Lamping, E. Mao, Distributed control for 3D metamorphosis. *Autonomous Robots* **10**, 41–56 (2001)
53. E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, S. Kokaji, A motion planning method for a self-reconfigurable modular robot, in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Springer, Berlin, Germany (2001)