

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Motion Planning and Reconfiguration for Systems of Multiple Objects

Adrian Dumitrescu¹

University of Wisconsin–Milwaukee
U.S.A.

1. Abstract

This chapter surveys some recent results on motion planning and reconfiguration for systems of multiple objects and for modular systems with applications in robotics.

1 Introduction

In this survey we discuss three related reconfiguration problems for systems of multiple objects.

(I) Consider a set of n pairwise disjoint objects in the plane that need to be brought from a given start (initial) configuration S into a desired goal (target) configuration T . The *motion planning* problem for such a system is that of computing a sequence of object motions (schedule) that achieves this task. If such a sequence of motions exists, we say that the problem is *feasible* and say that it is *infeasible* otherwise.

The problem is a simplified version of a multi-robot motion planning problem [23], in which a system of robots, whose footprints are, say disks, are operating in a common workplace. They have to move from their initial positions to a set of specified target positions. No obstacles other than the robots themselves are assumed to be present in the workplace; in particular, the workspace is assumed to extend throughout the entire plane. In many applications, the robots are indistinguishable so any of them can occupy any of the specified target positions; that is, the disks are unlabeled. Another application which 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 designing efficient algorithms for carrying out the motion plan. There are several types of moves, such as sliding or lifting, which lead to different models that will be discussed in Section 2.

(II) A different kind of reconfiguration problem appears in connection to so-called *self-reconfigurable* or *metamorphic* 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 [15, 11, 26, 27, 28, 32, 34, 35, 36, 39]. Typically, the modules have a regular symmetry so that they can be packed densely, with small gaps between them. Various practical realizations are under way at different sites. Such a system can be viewed as a large swarm of physically connected robotic modules that behave collectively as a single entity.

¹ Computer Science, University of Wisconsin–Milwaukee, Milwaukee, WI 53201-0784, USA. Email: ad@cs.uwm.edu. Supported in part by NSF CAREER grant CCF-0444188.

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 [10, 26, 38], or by expansion and contraction [32]. 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 [32]. 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 [18, 19, 20], focusing on two basic capabilities of such systems: reconfiguration and locomotion. We present details in Section 3.

(III) In many 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 target position. The case when the chips are labeled or unlabeled give 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 [13].

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 Section 4. We note that the three (disk) models mentioned earlier do not fall in the above graph reconfiguration framework, because a disk may partially overlap several target positions.

2 Models of reconfiguration for systems of objects in the plane

There are several types of moves that make sense to study, as dictated by specific applications, such as:

1. *Sliding model*: one move is sliding a disk to another location in the plane without intersecting any other disk, where the disk center moves along an arbitrary (open) continuous curve [5].
2. *Lifting model*: one move is lifting a disk and placing it back in the plane anywhere in the free space, that is, at a position where it does not intersect (the interior of) any other disk [6].
3. *Translation model*: one move is translating a disk in the plane along a fixed direction without intersecting any other disk [1].

It turns out that moving a set of objects from one place to another is related to certain *separability* problems [14, 9, 21, 22]; see also [31]. 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.

Model	Type	Lower bound	Upper bound
Translating disks	Congruent	$\lfloor 8n/5 \rfloor$	$2n - 1$
	Arbitrary	$2n$	$2n$
Sliding disks	Congruent	$16n/15 - o(n)$	$3n/2 + o(n)$
	Arbitrary	$2n - o(n)$	$2n - 1$
Lifting disks	Congruent	$n + \Omega(n^{1/2})$	$n + O(n^{2/3})$
	Arbitrary	$\lfloor 5n/3 \rfloor$	$9n/5$
Sliding chips (grid)	Unlabeled	n	n
	Labeled	$3n/2$	$7n/4$
Lifting chips (grid)	Unlabeled	n	n
	Labeled	$3n/2$	$3n/2$

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

It can be shown that for the class of disks, the reconfiguration problem in each of these models is always feasible [1, 5, 6, 9, 21, 22]. This follows essentially from the feasibility in the sliding model and the translation model, see Section 2.1. For the more general class of convex objects, one needs to allow rotations. For simplicity we will restrict ourselves mostly to the case of disks. We are thus lead 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?

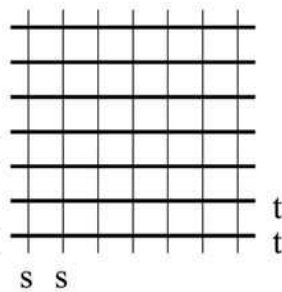


Fig. 1. $2n - 1$ moves are necessary (in either model) to bring the n segments from vertical position to horizontal position.

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 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 segments, as shown in Figure 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 mentioned can be found 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 Section 4. Table 1 which is constructed on the basis of the results in [1, 5, 6, 13] 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 appears in Figure 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. 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.

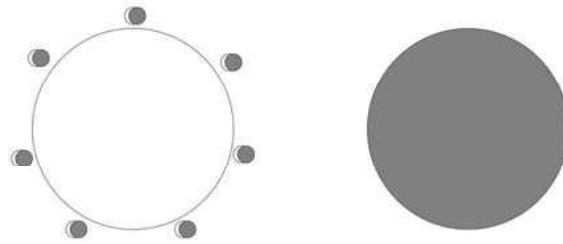


Fig. 2. One move is enough in the lifting model, while $n - 1$ are needed in the sliding model for this pair of start and target configurations with n disks each (here $n = 1$). Start disks are white and target disks are shaded.

A move is a *target move* if it moves a disk to a final target position. Otherwise, it is a *non-target* move. Our lower bounds use the following argument: if no target disk coincides with a start disk (so each disk must move), a schedule which requires x non-target 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 which follows from Theorem 1 below. This old result appears in the work of Fejes Tóth and Heppes [21], but it can be traced back to de Bruijn [9]; some algorithmic aspects of the problem have been addressed more recently by Guibas and Yao [22].

Theorem 1 [9, 21, 22] *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. Note 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 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.

Theorem 2 [5] *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 [5] 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 1 [5] *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 1. We can assume that ℓ is vertical. Denote by $s_1 = \lfloor n/2 \rfloor$ and $s_2 = \lfloor n/2 \rfloor$ the number of centers of start disks to the left and to the 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 to the 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 Figure 3. This illustration ignores the disks/targets in the center parallel strip.

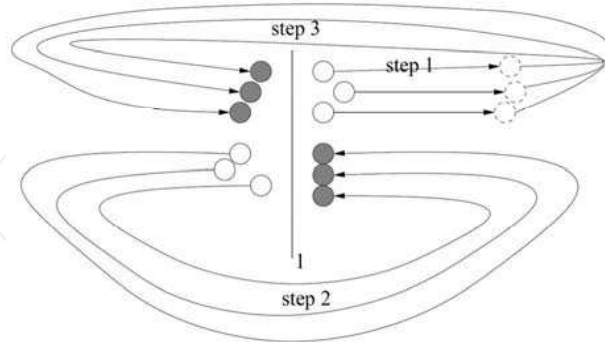


Fig. 3. Algorithm with three steps for sliding congruent disks. The start disks are white and the target disks are shaded.

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 Using all the $s'_1 \leq s_1 \leq n/2$ remaining disks whose centers are to the left of ℓ , in increasing order of their x -coordinates, fill free target positions whose centers are right of ℓ , in increasing order of their x -coordinates: each disk slides 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 non-target 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 $3n/2 + O(\sqrt{n \log n})$.

The first simple 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 Figure 4. Now observe that for each ring, the first move which involves a disk in that ring must be a non-target move. The number of rings is $\Theta(\sqrt{n})$, from which the lower bound $n + \Omega(\sqrt{n})$ follows.

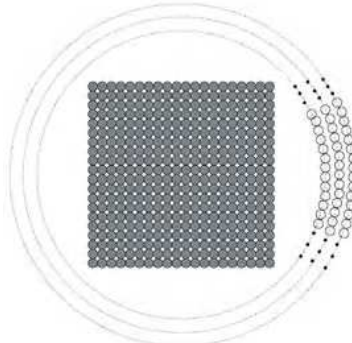


Fig. 4. A lower bound of $n + \Omega(\sqrt{n})$ moves: $\Theta(\sqrt{n})$ rings, each with $\Theta(\sqrt{n})$ start disk positions. Targets are densely packed in a square formation enclosed by the rings.

This basic idea of a cage-like construction can be further refined by redesigning the cage[5]. The new design is more complicated and uses “rigidity” considerations which go back to the stable disk packings of density 0 of K. Böröczky[7]. A packing C of unit (radius) disks in the plane is said to be *stable* if each disk is kept fixed by its neighbors [8]. More precisely, 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. Somewhat surprisingly, K. Böröczky [7] showed that there exist stable systems of unit disks with arbitrarily small density. These 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 shown in Figure 5.

Let us refer to the disks in the start (resp. target) configuration as white (resp. black) disks.

Now fix a large n , and take n 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. 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 neighbours. 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.

For disks of arbitrary radii, the following result is obtained in [5]:

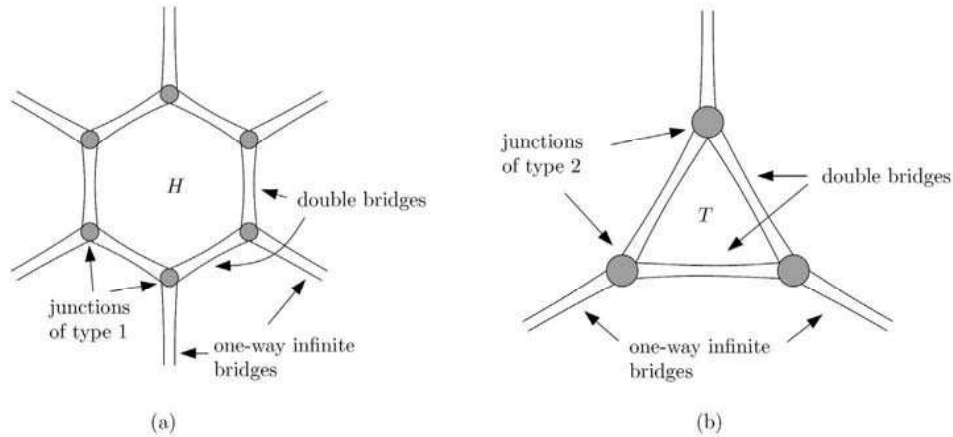


Fig. 5. Two start configurations based on hexagonal and triangular cage-like constructions. Targets are densely packed in a square formation enclosed by the cage.

For disks of arbitrary radii, the following result is obtained in [5]:

Theorem 3 [5] *Given a pair of start and target configurations, each consisting of n disks of arbitrary radii, $2n$ sliding moves always suffice for transforming the 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, for every sufficiently large n .*

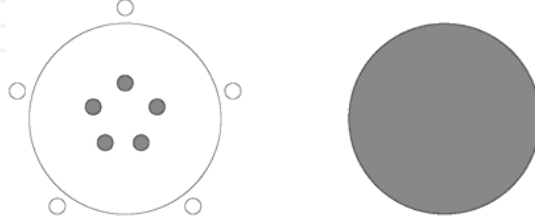


Fig. 6. A simple configuration which requires about $3n/2$ moves (basic step for the recursive construction).

The upper bound follows from the universal reconfiguration algorithm described earlier. The lower bound is a recursive construction shown in Figure 7. It is obtained by iterating recursively the basic construction in Figure 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 $3n/2$ moves in total are necessary.

In the recursive construction, the small disks around a large one are replaced by the "same" construction scaled (see Figure 7). All disks have distinct radii so it is convenient to think of them as being labeled. There is one large disk labeled 0, 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 levels in the recursion, about $n/2 + n/4 + \dots + n/2^k$ non-target 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 [5].

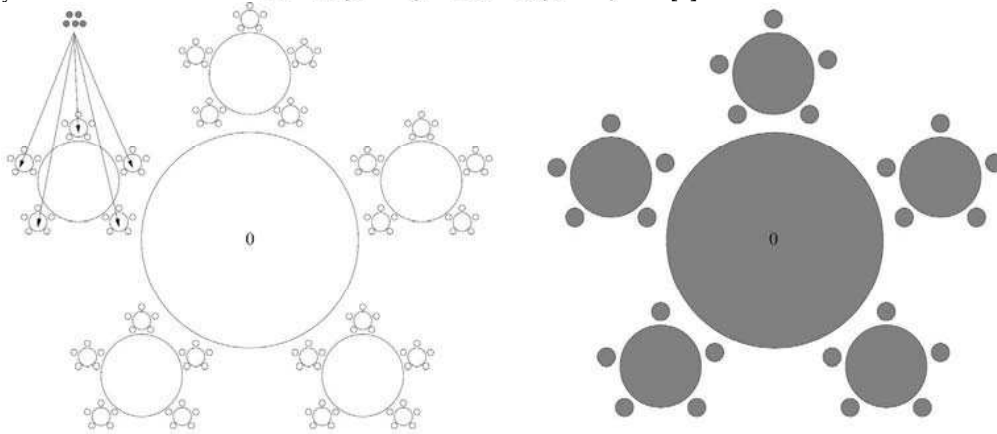


Fig. 7. Recursive lower bound construction for sliding disks of arbitrary radii: $m = 2$ and $k = 3$.

2.2 The translation model

This model 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 all almost parallel to any given fixed direction using $2n$ translation moves [1].

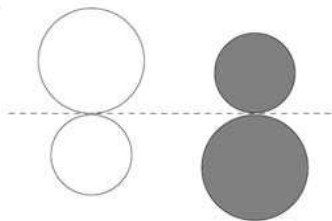


Fig. 8. A two-disk configuration that requires 4 translation moves.

That this bound is best possible for arbitrary radii disks can be easily seen in Figure 8. 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 non-target 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.

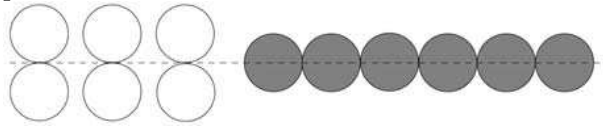


Fig. 9. A configuration of 6 congruent disks that requires 9 translation moves.

Theorem 4 [1] *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 Figure 9 requires $3n/2$ moves, since from each pair of tangent disks, the first move must be a non-target move. The best known lower bound, $\lceil 8n/5 \rceil$, is from [1]; we illustrate it in Figure 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. There are essentially only two different ways for “moving” one group, each requiring three non-target moves: (i) 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. (ii) S_4 , S_5 and S_2 move out (to the left), S_1 and S_3 move to destinations followed by the rest.

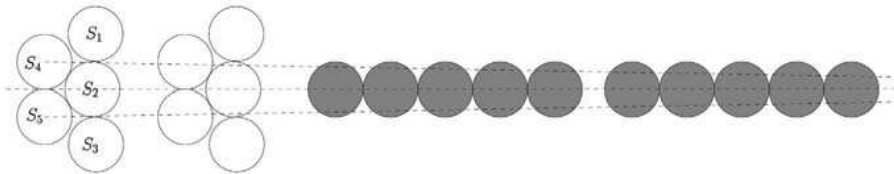


Fig. 10. Reconfiguration of a system of 10 congruent disks which needs 16 translation moves. Start disks are white and target disks are shaded.

Theorem 5 [1] *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 $\lceil 8n/5 \rceil$ such moves.*

2.3 The lifting model

For systems of n congruent disks, one can estimate the number of moves that are always sufficient with relatively higher accuracy. The following result is obtained in [6]:

Theorem 6 [6] *Given a pair of start and target configurations S and T , each with n congruent disks, one can move 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*

which require $n + \Omega(n^{1/2})$ moves for this task.

The lower bound construction is illustrated in Figure 11 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 [6], at least $n + \lceil m/2 \rceil$ moves are necessary for reconfiguration (it can be verified that this number of lifting moves suffices for this construction).

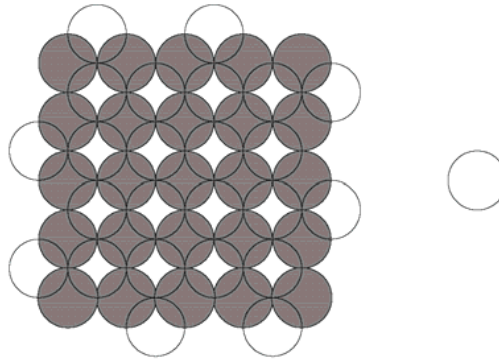


Fig. 11. A pair of start and target configurations, each with $n = 25$ congruent disks, which require 27 lifting moves. The start disks are white and the target disks are shaded.

The upper bound $n + O(n^{2/3})$ of is technically somewhat more complicated. It uses a binary space partition of the plane into convex polygonal (bounded or 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 [6] 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 which performs a number of moves close to the optimum (for large n).

For arbitrary radius disks, the following result is obtained in [6]:

Theorem 7 [6] *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 which require $\lceil 5n/3 \rceil$ 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 Figure 12. 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 non-target 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 $\lceil 5n/3 \rceil$ in general.

We now explain the approach in [6] for the upper bound for 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.

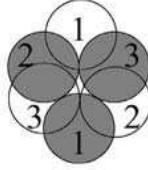


Fig. 12. A group of three disks which require five moves to reach their targets; part of the lower bound construction for lifting disks of arbitrary radii. The disks are white and their targets are shades.

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 [6] 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 non-target 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 far away 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 far away disks in any order. The number of moves is at most

$n + 4n/5 = 9n/5$, as claimed. Figure 13 shows the bipartite intersection graph G and the directed blocking graph D for a small example, with the corresponding reconfiguration procedure explained above. 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 (with ratio $9/5$).

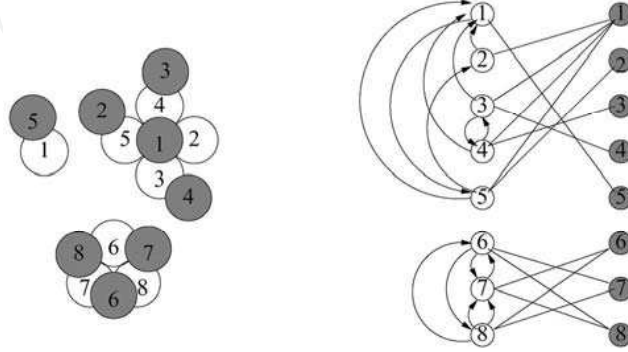


Fig. 13. The bipartite intersection graph G and the directed blocking graph D . Move out: 4, 5, 7, 8; no cycles remain in D . Fill targets: 3, 2, 1, 6, and then 4, 5, 7, 8. The start disks are white and the target disks are shaded.

2.4 Further questions

Here are some interesting remaining questions pertaining to the sliding and translation models:

- (1) 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 (Figure 11) 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?
- (2) Derive a $(2 - \delta)n$ upper bound for the case of congruent disks in the translation model (where δ is a positive constant), or improve the $\lfloor 8n/5 \rfloor$ lower bound.
- (3) Consider the reconfiguration problem for congruent *labeled* disks in the sliding model. It is easy to see that $\lfloor 5n/3 \rfloor$ the lower bound for arbitrary disks holds, since the construction in Figure 12 can be realized with congruent disks. Find a $(2 - \delta)n$ upper bound (where δ is a positive constant), or improve the $\lfloor 5n/3 \rfloor$ lower bound.

3. Modular and reconfigurable systems

In [20] a number of issues related to motion planning and analysis of rectangular metamorphic robotic systems are addressed. 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 were also addressed. In particular the following two questions have been shown to be decidable: (i) whether a given set of motion rules maintains connectivity; (ii) 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 [18, 19, 20]. Consider a plane that is partitioned into a 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: the module-size determines the grid size in practice, and similarly for orientation.

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 (distributions) 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 we often identify V_t with the set of cells occupied by the modules or with the set of their centers. We are only interested in configurations that are connected, 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 (Figure 14) define the *rectangular model* [18, 19, 20]. These are to be understood as possible in all axis parallel orientations, in fact generating 16 rules, eight for rotation and eight for sliding.

- *Rotation*: A module m side-adjacent to a stationary module f rotates through an angle of 90° around f either clockwise or counterclockwise. Figure 14(a) 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 that m 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 and side-adjacent to f_2 . Figure 14(b) 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). Parallel execution has the advantage to speed up the reconfiguration process. If concurrent moves are allowed, additional conditions have to be imposed, as in [19, 20]. In order to ensure motion precision, each move is guided by one or two modules that are stationary during the same step.

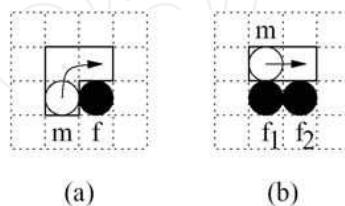


Fig. 14. Moves in the rectangular model: (a) clockwise *NE* rotation and (b) sliding in the *E* direction. Fixed modules are shaded. The cells in which the moves take place are outlined in the figure.

The following recent result settles a conjecture formulated in [20].

Theorem 8 [18] *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 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, for sequential execution) are quite involved. At the moment, this is for reconfiguration without the presence of obstacles!

We refer to a set of modules that form a straight line chain in the grid, as a *straight chain*. It is easy to construct examples so that neither sliding nor rotation alone can reconfigure them to straight chains. Conform with Theorem 8, the motion rules of the rectangular model (rotation and sliding, Figure 14) are sufficient to guarantee reachability, while maintaining the system connected at each discrete time step. This has been proved earlier for the special class of horizontally convex systems [20].

A somewhat different model can be obtained if, instead of the connectedness requirement at each time step, one imposes the so-called *single backbone* condition [20]: a module moves (slides or rotates) along a single connected backbone (formed by the other modules). If concurrent moves are allowed, additional conditions have to be imposed, as in [20]. 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 Figure 15: the initial connected configuration practically disconnects during the move and reconnects at the end of it. Our algorithm has the nice property that the single backbone condition is satisfied during the whole procedure.

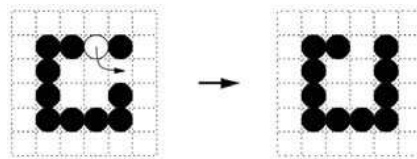


Fig. 15. A rotation move which temporarily disconnects the configuration.

We now briefly discuss another rectangular model for which the same property holds. The following two generic motion rules (Figure 16) define the *weak rectangular model*. These are to be understood as possible in all axis-parallel orientations, in fact generating eight rules, four diagonal moves and four side moves (axis-parallel ones). The only imposed condition is that the configuration 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 $cell(m)$.

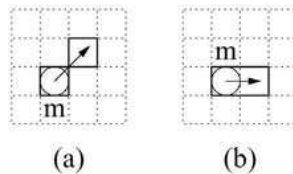


Fig. 16. Moves in the weak rectangular model: (a) NE diagonal move and (b) side move in the E direction. The cells in which the moves take place are outlined in the figure.

The same result as in Theorem 8 holds for this second model [18]; however, its proof and corresponding reconfiguration algorithm, are much simpler. It remains to be seen how these two models compare in a practical realization.

Theorem 9 [18] *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.*

A different variant of inter-robot 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 [12]. 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 which are to explore in parallel a large area. Once the task is complete, the robots might merge back into the large robot that carried them.

As mentioned in [17], there is considerable research interest in the task of having one autonomous vehicle follow another, and in general in studying robots moving in formation. [19] examines 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 in the rectangular model, for the case of even $n \geq 4$, there exist snake-like formations having a speed of $\frac{1}{3}$. Fig. 17 shows the case $n = 20$, where 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 $\frac{1}{3}$.

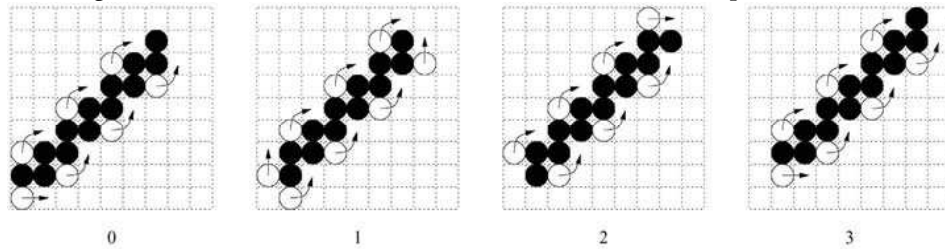


Fig. 17. Formation of 20 modules moving diagonally at a speed of $\frac{1}{3}$ (diagonal formation).

We conclude this section with some remaining questions on modular and reconfigurable systems related to the results presented:

- (1) The reconfiguration algorithm in the rectangular model takes fewer than $2n^3$ moves in the worst case (with the current analysis). 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. This has been shown to hold in the weak rectangular model, but it remains open in the first model.
- (2) Extend the algorithm(s), for reconfiguration under the presence of obstacles.
- (3) 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.

4 Reconfigurations in graphs and grids

In certain applications, objects are indistinguishable, therefore the chips are unlabeled; for instance, a modular robotic system consists of a number of identical modules (robots), each having identical capabilities [18, 19, 20]. 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 [24, 33]. Most of the work done so far concerns labeled versions of the reconfiguration problem, and we give here 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 [37] 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 [25]. Ratner and Warmuth have shown that finding a solution with minimum number of moves for the $(N \times N)$ extension of the 15-puzzle is intractable [30], 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* [3]. This problem is the labeled variant of the same reconfiguration problem studied in [13], 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 s , that has to reach to a designated vertex t using the smallest number of moves, in the presence of obstacles (pebbles) at some of the other vertices [29]. 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 ratio $O(\sqrt{n})$ polynomial time approximation algorithm was given in [29].

The following results are shown in [13] for the “chips in graph” reconfiguration problem described in part (III) of Section 1.

- (1) The reconfiguration problem in graphs with unlabeled chips U-GRAPH-RP is NP-hard, and even APX-hard.
- (2) The reconfiguration problem in graphs with labeled chips L-GRAPH-RP is APX-hard.
- (3) For the infinite planar rectangular grid, both the labeled and unlabeled variants L-GRID-RP and U-GRID-RP are NP-hard.
- (4) There exists 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.
- (5) It is 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 which performs an optimal (minimum) number of moves is presented.
- (6) 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 10 [13] *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 is sufficient to prove the theorem for trees. 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 . Move first 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 10 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 [5], as discussed in Section 2.1. Here are some interesting remaining questions on reconfigurations in graphs and grids:

- (1) Can the ratio 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) 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 [18] were derived from the properties of a system of *maximal cycles*, similar to those of the block decomposition of graphs [16].

The lower bound configuration with unit disks for the sliding model in [5] uses “rigidity” considerations and properties of *stable* packings of disks studied a long time ago by Böröczky [7]; in particular, he showed that there exist 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 [6] is intimately related to the notion of *center point* of a finite point set, and to the following fact 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 [6], similar to the bound on the *independence number* of an undirected graph given by Turán's theorem [2].

The ratio 3 approximation algorithm for the unlabeled version in graphs is obtained by applying the local ratio method of Bar-Yehuda [4] to a graph H constructed from the given graph G . Regarding the various models of reconfiguration for systems of objects in the plane, we currently have combinatorial 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.

6 References

- [1] M. Abellanas, S. Bereg, F. Hurtado, A. G. Olaverri, D. Rappaport, and J. Tejel, Moving coins, *Computational Geometry: Theory and Applications*, 34(1), 2006, 35–48.
- [2] N. Alon and J. Spencer, *The Probabilistic Method*, second edition, Wiley, New York, 2000.
- [3] V. Auletta, A. Monti, M. Parente, and P. Persiano, A linear-time algorithm for the feasibility of pebble motion in trees, *Algorithmica*, 23 (1999), 223–245.
- [4] R. Bar-Yehuda, One for the price of two: a unified approach for approximating covering problems, *Algorithmica*, 27 (2000), 131–144.
- [5] S. Bereg, A. Dumitrescu, and J. Pach, Sliding disks in the plane, *Proceedings of Japan Conference on Discrete and Computational Geometry 2004* (J. Akiyama, M. Kano and X. Tan, editors), LNCS vol. 3742, Springer, 2005, 37–47.
- [6] S. Bereg and A. Dumitrescu, The lifting model for reconfiguration, *Discrete & Computational Geometry*, accepted. A preliminary version in *Proceedings of the 21st Annual Symposium on Computational Geometry*, (SOCG '05), Pisa, Italy, June 2005, 55–62.
- [7] K. Böröczky, Über stabile Kreis-und Kugelsysteme (in German), *Ann. Univ. Sci. Budapest. Eötvösect. Math.* 7 (1964), 79–82.
- [8] P. Braß, W. Moser, and J. Pach, *Research Problems in Discrete Geometry*, Springer, New York, 2005.
- [9] N. G. de Bruijn, Aufgaben 17 and 18 (in Dutch), *Nieuw Archief voor Wiskunde* 2 (1954), 67.
- [10] Z. Butler, K. Kotay, D. Rus and K. Tomita, Generic decentralized control for a class of selfreconfigurable robots, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA '02)*, Washington, May 2002, 809–816.
- [11] A. Casal and M. Yim, Self-reconfiguration planning for a class of modular robots, *Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing*, (SPIE'99), 246–256.
- [12] A. Castano and P. Will, A polymorphic robot team, in *Robot Teams: From diversity to Polymorphism*, edited by T. Balch and L.E. Parker, pp. 139–160, A K Peters, 2002.
- [13] G. Călinescu, A. Dumitrescu, and J. Pach, Reconfigurations in graphs and grids, *Proceedings of Latin American Theoretical Informatics Symposium (LATIN '06)*,

- Valdivia, Chile, March 2006, LNCS vol. 3887, Springer, 2006, 262–273.
- [14] B. Chazelle, T. A. Ottmann, E. Soisalon-Soininen, and D. Wood, The complexity and decidability of separation, *Proceedings of the 11th International Colloquium on Automata, Languages and Programming*, (ICALP '84), LNCS vol. 172, Springer, 1984, pp. 119–127.
 - [15] G. Chirikjian, A. Pamecha and I. Ebert-Uphoff, Evaluating efficiency of self-reconfiguration in a class of modular robots, *Journal of Robotic Systems*, 13(5) (1996), 317–338.
 - [16] R. Diestel, *Graph theory (2nd ed.)*, *Graduate Texts in Mathematics* 173, Springer-Verlag, New York, 2000.
 - [17] G. Dudek, M. Jenkin, and E. Milios, A taxonomy of multirobot systems, in *Robot Teams: From diversity to Polymorphism*, edited by T. Balch and L.E. Parker, pp. 3–22, A K Peters, 2002.
 - [18] A. Dumitrescu and J. Pach, Pushing squares around, *Graphs and Combinatorics*, 22(1) (2006), 37–50.
 - [19] A. Dumitrescu, I. Suzuki, and M. Yamashita, Motion planning for metamorphic systems: feasibility, decidability and distributed reconfiguration, *IEEE Transactions on Robotics and Automation*, 20(3), 2004, 409–418.
 - [20] A. Dumitrescu, I. Suzuki, and M. Yamashita, Formations for fast locomotion of metamorphic robotic systems, *International Journal of Robotics Research*, 23(6), 2004, 583–593.
 - [21] L. Fejes Tóth and A. Heppes, Über stabile Körper Systeme (in German), *Compositio Mathematica*, 15 (1963), 119–126.
 - [22] L. Guibas and F. F. Yao, On translating a set of rectangles, in *Computational Geometry*, F. Preparata (ed.), pp. 61–67, Vol. 1 of *Advances in Computing Research*, JAI Press, London, 1983.
 - [23] Y. Hwuang and N. Ahuja, Gross motion planning — a survey, *ACM Computing Surveys*, 25(3) (1992), 219–291.
 - [24] W. W. Johnson, Notes on the 15-puzzle. I., *American Journal of Mathematics*, 2 (1879), 397–399.
 - [25] D. Kornhauser, G. Miller, and P. Spirakis, Coordinating pebble motion on graphs, the diameter of permutation groups, and applications, *Proceedings of the 25-th Symposium on Foundations of Computer Science*, (FOCS '84), 241–250.
 - [26] S. Murata, H. Kurokawa and S. Kokaji, Self-assembling machine, *Proceedings of IEEE International Conference on Robotics and Automation*, (1994), 441–448.
 - [27] A. Nguyen, L. J. Guibas and M. Yim, Controlled module density helps reconfiguration planning, *Proceedings of IEEE International Workshop on Algorithmic Foundations of Robotics*, 2000.
 - [28] A. Pamecha, I. Ebert-Uphoff and G. Chirikjian, Useful metrics for modular robot motion planning, *IEEE Transactions on Robotics and Automation*, 13(4) (1997), 531–545.
 - [29] C. Papadimitriou, P. Raghavan, M. Sudan, and H. Tamaki, Motion planning on a graph, *Proceedings of the 35-th Symposium on Foundations of Computer Science*, (FOCS '94), 511–520.
 - [30] D. Ratner and M. Warmuth, Finding a shortest solution for the $(N \times N)$ extension of the 15-puzzle is intractable, *Journal of Symbolic Computation*, 10 (1990), 111–137.
 - [31] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, second

- edition, 1998.
- [32] D. Rus and M. Vona, Crystalline robots: self-reconfiguration with compressible unit modules, *Autonomous Robots*, 10 (2001), 107–124.
 - [33] W. E. Story, Notes on the 15 puzzle. II., *American Journal of Mathematics*, 2 (1879), 399–404.
 - [34] J.E. Walter, J.L. Welch and N.M. Amato, Distributed reconfiguration of metamorphic robot chains, *Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC'00)*, July 2000, 171–180.
 - [35] J.E. Walter, J.L. Welch and N.M. Amato, Reconfiguration of hexagonal metamorphic robots in two dimensions, in *Sensor Fusion and Decentralized Control in Robotic Systems III* (G. T. McKee, P. S. Schenker, editors), *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, Vol. 4196, 2000, 441–453.
 - [36] X. Wei, W. Shi-gang, J. Huai-wei, Z. Zhi-zhou, and H. Lin, Strategies and methods of constructing self-organizing metamorphic robots, in *Mobile Robots: New Research*, edited by J. X. Liu, pp. 1–38, Nova Science, 2005.
 - [37] R. M. Wilson, Graph puzzles, homotopy, and the alternating group, *Journal of Combinatorial Theory, Series B*, 16 (1974), 86–96.
 - [38] M. Yim, Y. Zhang, J. Lamping and E. Mao, Distributed control for 3D metamorphosis, *Autonomous Robots*, 10 (2001), 41–56.
 - [39] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa and 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*, 2001.



Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

ISBN 3-86611-283-1

Hard cover, 704 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, February, 2007

Published in print edition February, 2007

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well as those handling global navigation aspects of a single robot and even for a group of robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Adrian Dumitrescu (2007). Motion Planning and Reconfiguration for Systems of Multiple Objects, Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), ISBN: 3-86611-283-1, InTech, Available from: http://www.intechopen.com/books/mobile_robots_perception_navigation/motion_planning_and_reconfiguration_for_systems_of_multiple_objects

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen