Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots

Gregory Chirikjian*
Amit Pamecha
Imme Ebert-Uphoff
Department of Mechanical Engineering
The Johns Hopkins University
Baltimore, MD 21218

Received May 4, 1995; accepted July 25, 1995

In this article we examine the problem of dynamic self-reconfiguration of a class of modular robotic systems referred to as *metamorphic* systems. A metamorphic robotic system is a collection of mechatronic modules, each of which has the ability to connect, disconnect, and climb over adjacent modules. A change in the macroscopic morphology results from the locomotion of each module over its neighbors. Metamorphic systems can therefore be viewed as a large swarm of physically connected robotic modules that collectively act as a single entity. What distinguishes metamorphic systems from other reconfigurable robots is that they possess all of the following properties: (1) a large number of homogeneous modules; (2) a geometry such that modules fit within a regular lattice; (3) self-reconfigurability without outside help; (4) physical constraints which ensure contact between modules. In this article, the kinematic constraints governing metamorphic robot self-reconfiguration are addressed, and lower and upper bounds are established for the minimal number of moves needed to change such systems from any initial to any final specified configuration. These bounds are functions of initial and final configuration geometry and can be computed very quickly, while it appears that solving for the precise number of minimal moves cannot be done in polynomial time. It is then shown how the bounds developed here are useful in evaluating the performance of heuristic motion planning/reconfiguration algorithms for metamorphic systems.

*To whom all correspondence should be addressed

Journal of Robotic Systems 13(5) 317–338 (1996)

CCC 0741-2223/96/050317-22

この発表では、Metamorphic システムと呼ばれるモジュラ・ロボット・システムのクラスに対する力学的自己再配置問題について考察する。Metamorphic ロボット・システムは、メカトロニック・モジュールの集合であり、各々は連結、切断、隣接モジュールの乗り越えができる。巨視的形態論における変化は、各モジュールと隣接モジュールの移動によって起こる。そして Metamorphic システムは、物理的に連結されたロボット・モジュールの大きな集団に見え、集合体は1つの構造体として動作する。他の再配置可能なロボットと Metamorphic システムの違いは、次の属性を持つことである。(1) 多数の均質モジュール、(2) 規則正しい格子の中に収まるモジュールの幾何学構造 (3) 外部からの援助に頼らない自己再配置能力 (4) モジュール間の接触を確実にする物理的強制。
この発表では、Metamorphic ロボットの自己再配置を支配する運動学的制約について説明し、このようなシステムが任意の幾何学な初期配置から最終配置に変化するのに必要な最少の移動回数に対する上限と下限を高速で計算して設定する。一方、多項時間の中で正確な最少移動の回数を求めるのは困難と思われる。そして、ここで展開される範囲が、Metamorphic システムにおいて試行錯誤による動作計画／再配置アルゴリズムの性能評価に有効であることを示す。

1. INTRODUCTION

A *metamorphic* robotic system is a collection of independently controlled mechatronic modules, each of which has the ability to connect, disconnect, and climb over adjacent modules. Each module allows power and information to flow through itself and to its neighbors. A change in the metamorphic manipulator morphology (i.e., a change in the relative location of modules within the collection) results from the locomotion of each module over its neighbors. Thus a metamorphic system has the ability to dynamically self-reconfigure. Changes in configuration within a given morphology are achieved by changing joint angles, as is the case for standard (fixed-morphology) robotic manipulators.

Metamorphic systems can therefore be viewed as a large swarm (or colony) of connected robots that collectively act as a single entity. What distinguishes metamorphic systems from other reconfigurable robots is that they possess all of the following properties:

1. All modules have the same physical structure, and each must have complete computational and communications functionality. In this way uniform treatment of modules in the planning problem is possible.
2. Symmetries in the mechanical structure of the modules must be such that they fill planar and spatial regions with minimal gaps. In this way, a lattice of modules is formed for any task.
3. The modules must each be kinematically sufficient with respect to the task of locomotion, i.e., they must have enough degrees of freedom to be able to "walk" over adjacent modules so that they can reconfigure without outside help.
4. Modules must adhere to adjacent modules, e.g., there must be electromechanical or electromagnetic connectors between modules that can carry load. In this way the collection of modules becomes a single physical object.

Although a wide variety of module designs satisfy the above conditions, one particular class is discussed here. These modules are mechanisms that can be represented as polyhedra, e.g., certain kinds of platform manipulators in the spatial case, or closed linkages in the plane. Figures 1 and 2 show a planar example where the modules are six bar linkages. Hardware implementations constructed by other authors that satisfy the above conditions can be found in the literature.[1-3] We have built metamorphic robot modules capable of self-reconfiguration as shown in the sequence of hardware photographs in Figure 2(b). For details regarding the connector mechanism design and other aspects of the hardware implementation, see Pamecha et al.[4] and Pamecha and Chirikjian.[5]

To satisfy condition (1) above, regular polyhedral module designs are assumed, e.g., closed loop mechanisms with uniform link lengths,[6] although other designs do fit within this framework. In this way, the modules are not only uniform, but also

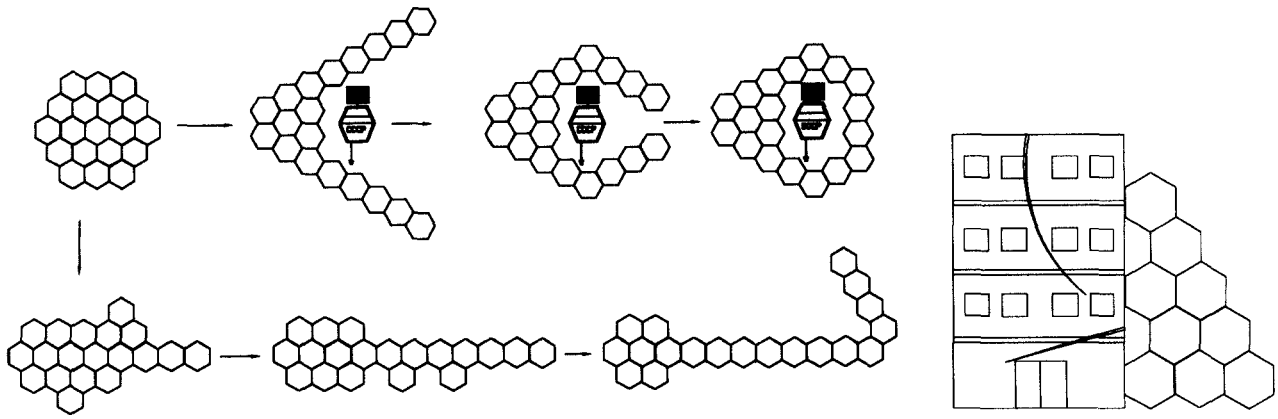


Figure 1. Examples of metamorphic robot applications.

possess a multiplicity of rotational symmetries. Condition (2) is then reduced to selecting regular polyhedra that close-pack (or nearly close-pack) space.

Potential applications of metamorphic systems include: (1) obstacle avoidance in highly constrained and unstructured environments; (2) "growing" structures composed of modules to form bridges, buttresses, and other civil structures in times of emergency; (3) envelopment of objects, such as recovering satellites from space. Some of these applications are shown in Figure 1.

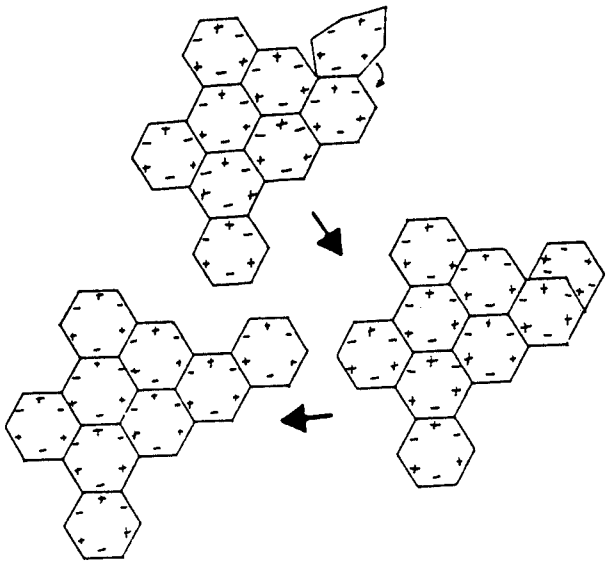
This article addresses issues in the kinematics and motion planning of metamorphic systems with a fixed base, i.e., "manipulators," as opposed to "mobile robots." No distinction is made between "motion planning" and "self-reconfiguration" of these systems—these words are synonymous in the context of metamorphic systems. In section 2, a brief review of the related literature is presented. In section 3, the kinematic and metric properties associated with metamorphic systems are examined for the general case. In section 4, we formalize the motion planning/reconfiguration problem and establish lower and upper bounds on the number of moves required to reconfigure between any given initial and final configurations. Section 5 presents a method for improving the performance of a given move sequence based on the bounds developed in section 4. Section 6 presents examples of this general formulation.

2. LITERATURE REVIEW

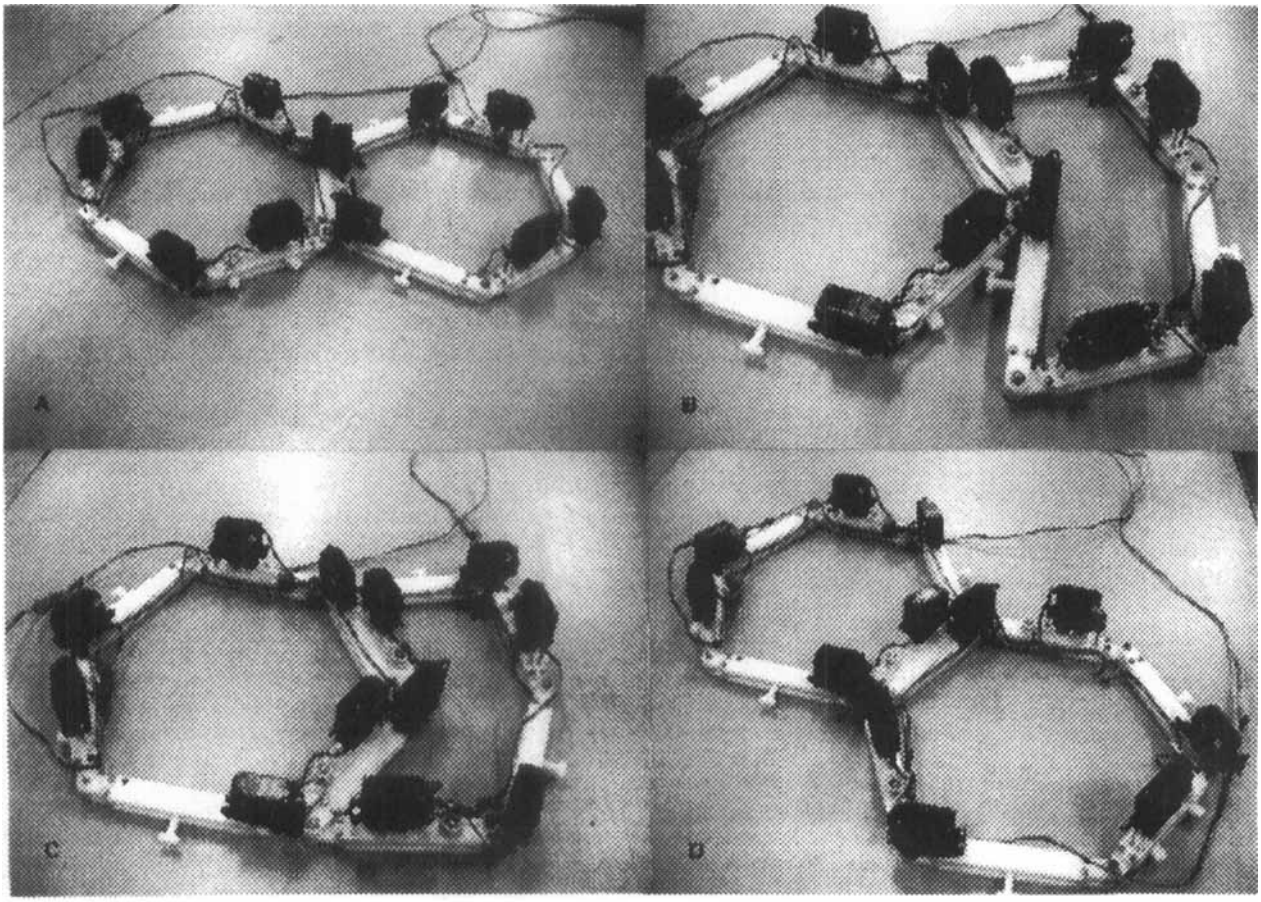
The idea of a metamorphic robotic system differs from related concepts presented in the literature. Three types of modular reconfigurable robotic systems have been proposed in the literature: (1) robots in which modules are reconfigured using external intervention[7–11]; (2) cellular robotic systems in which a heterogeneous collection of independent specialized modules are coordinated[12–17]; and (3) swarm intelligence in which there are generally no physical connections between modules.[18–20] Most recently, two other types of modular reconfigurable robotic systems have been considered. Yim[2,3] considers modular robots composed of a few basic elements that can be composed into complex systems and used for various modes of locomotion. Murata et al.[1] examine a "fractal" system composed of modules with zero kinematic mobility, but that can "walk" over each other in discrete quanta due to changes in the polarity of magnetic fields. A valuable tool for defining equivalence classes of modular robot configurations with the same shape and morphological function is provided in Chen and Burdick[21] and Chen.[22] Configurations within each of these classes are said to be isomorphic to each other.

The concept of a metamorphic system differs from concepts in the related works mentioned above because modules are homogeneous in form and function, physical contact between modules must always occur, self-reconfiguration is possible, and the resulting structures have the ability to act as manipulators because each module has full kinematic mobility. Nonetheless, the methods developed in this article are applicable to other types of self-reconfigurable systems. For instance, the "fractal" modules introduced in Murata et al.[1] exhibit all but the mobility requirement, and thus many of the methods developed in this article for kinematics and motion planning apply.

As the number of modules in a metamorphic



(a)



(b)

Figure 2. The reconfiguration process: (a) illustration; (b) hardware demonstration.

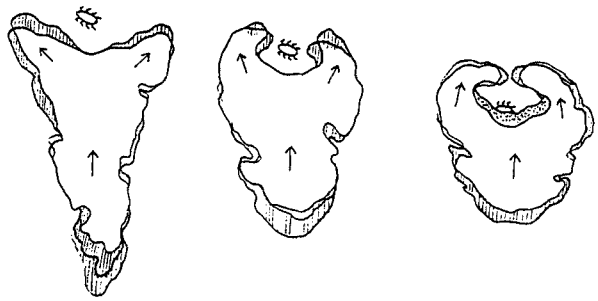


Figure 3. An amoeba.

system becomes very large, the manipulator could be viewed as a "mechatronic amoeba" because the manipulator takes on a continuous appearance. Figure 3 illustrates an amoeba. Figure 4 shows how a slime mold can reconfigure itself. Thus, the idea of metamorphic structures is not foreign to the natural world. For further reading on these subjects see Jeon,[23] Schaeffer,[24] and Bonner.[25]

In the next section we formalize the kinematic issues pertaining to metamorphic robots.

3. LATTICE KINEMATICS

This section addresses issues in the lattice kinematics of metamorphic robotic systems. There are three separate issues pertaining to the kinematics of a metamorphic system. First, there is the *continuous* motion problem of module kinematics, which is the time-evolution of joint angles required to execute the locomotion of a module from any given position to an adjacent one. This is illustrated in Figure 2 for the planar case with hexagonal modules, and analyzed in Chirikjian.[6] Second, there is the description of macroscopic configuration. This is a *discrete* problem that we call the lattice kinematics problem, and that is the focus of this section. Third, there are forward and inverse kinematics issues pertaining to each metamorphic robot configuration when it is being used as a fixed morphology manipulator, which can be addressed using existing redundancy resolution or motion planning techniques, e.g., Chirikjian[26] (and references therein).

In subsection 3.1 a general lattice coordinate system for describing macroscopic configurations is presented and associated metric properties are enumerated. That is, useful methods of measuring distance between *modules* within a lattice are discussed. In subsection 3.2, the metric properties of the set of all connected combinations of modules fixed to a common base are examined, i.e., we develop concepts of how distant two *configurations* with the same number of modules are within the space of all configurations.

3.1. Defining Distance Between Modules

Consider $\mathbf{R}^N$ where $N = 2$ or 3. A *lattice* is a discrete subset of $\mathbf{R}^N$ defined by a set of N linearly independent unit vectors, $\vec{v}_i \in \mathbf{R}^N$, as follows:

$$\mathbf{L}^N = \left\{ \sum_{i=1}^{N} k_i \vec{v}_i : k_1, k_2, \ldots, k_N \in \mathbf{Z} \right\}.$$

A vast body of literature deals with the symmetry groups associated with lattices (which are simply discrete subgroups of E(N)—the Euclidean Group), and the decomposition of space into polyhedra whose centers form regular lattices, e.g., Gericke[27] and Miller.[28]

In our problem, elements of the lattice (individual polyhedral cells) are either filled with robotic modules or obstacles or remain empty. $\mathbf{R}^N$ is then viewed as a lattice of regular polyhedra that are either empty or filled. One way to coordinatize the lattice and represent the relative positions of the centers of each polyhedral cell is to use the *Miller-Bravais* framework used in materials science and crystallography.[29] In this framework, a redundant set of coordinates are often used to describe the position of each point. An alternate framework is simply to use the vectors $\{\vec{v}_i\}$ to define coordinate axes. This is illustrated in Figure 5(a) for a hexagonal lattice. By denoting the origin as the vector $\vec{0} \in \mathbf{R}^N$ centered at the fixed base module, and defining unit vectors along any N independent directions that contain at least two lattice points (module centers), every point in the lattice is given a unique set of coordinates with the above described unit vectors defining coordinate axes. While this coordinate system will generally be skewed, it is much more convenient than an orthogonal system for describing the relative location of modules within the lattice.

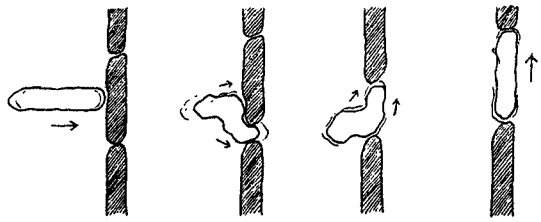


Figure 4. A slime mold.

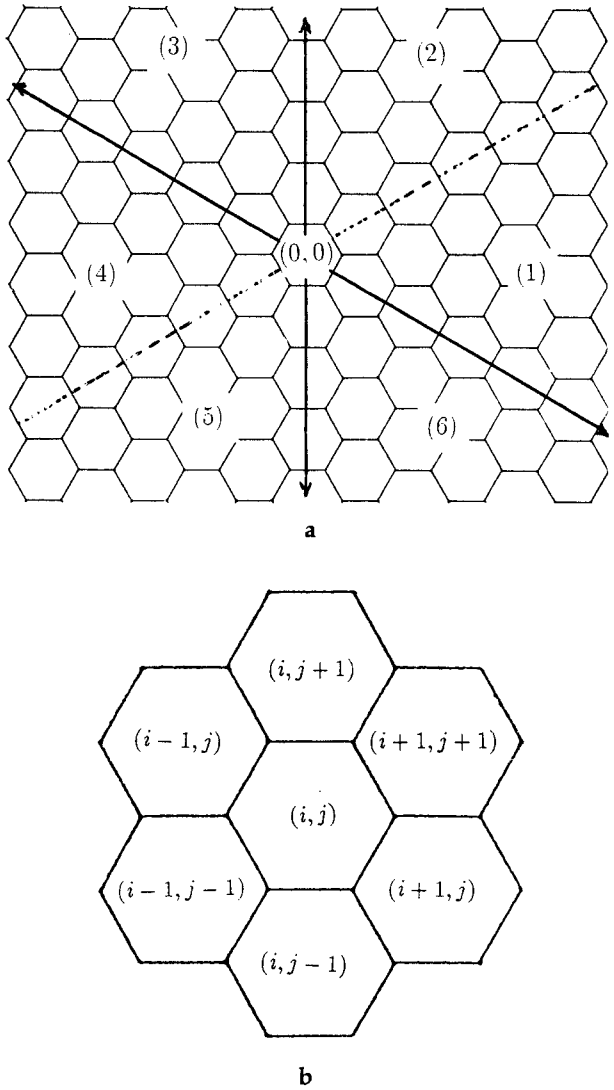


Figure 5. (a) Coordinate axes in a hexagonal lattice; (b) generating patch.

Figure 5(b) shows how coordinates of neighboring lattice points can be generated iteratively by translating a discrete coordinate patch.

For the kinematics of the lattice to be complete, the most important geometric quantity of all must be defined in our coordinate system, namely: distance. A proper distance (or metric) function between points a and b in any given set will satisfy the following properties[30]:

$$\begin{aligned} &\delta(a, b) \geq 0 \text{ and } \delta(a, b) = 0 \text{ iff } a = b \\ &\delta(a, b) = \delta(b, a) \\ &\delta(a, b) + \delta(b, c) \geq \delta(a, c), \end{aligned} \tag{1}$$

which we refer to as positive definiteness, symmetry, and the triangle inequality, respectively. The original set, together with a metric function defined on that set is called a *metric space.*

Since the sets we will be dealing with are lattices in $\mathbf{R}^N$, each module center will be represented by N independent integers, and we will make no distinction between a point in a lattice, and the coordinates that describe the point. This is, if $\vec{a}$ represents point a then $a = \vec{a} = (a_1, \ldots, a_N)^T$. Likewise, $b = \vec{b} = (b_1, \ldots, b_N)^T$ if $\vec{b}$ represents the point b. Since the point set we will be using is the set of all the centers of given polyhedra that form a close-packed (or nearly close-packed) lattice, this is a discrete subset of $\mathbf{R}^N$, and so any metric defined in $\mathbf{R}^N$ will be a metric on this set. We will call elements of this set *lattice points.*

One commonly used metric for $\mathbf{R}^N$ when using Cartesian coordinates is the "Taxicab"[31] (also called the Manhattan[32]) metric:

$$\delta_1(a, b) = \delta_1(\vec{a}, \vec{b}) = \sum_{i=1}^{N} |a_i - b_i|, \tag{2}$$

where $|\cdot|$ is the absolute value of a real number. This metric gets its name from the distance that one must travel in an ideal city in which every block is the same size and every street is perpendicular to every other. Note that the minimally distant paths between two points using this metric are generally not unique, with the exception of points that lie on a line parallel to a coordinate axis. This will also be the case for the metric to be discussed shortly, which is a generalization of the taxicab metric to nonrectangular lattices.

Another commonly used metric for $\mathbf{R}^N$ when using Cartesian coordinates is the Euclidean metric:

$$\delta_2(a, b) = \delta_2(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}. \tag{3}$$

This metric defines the length of the unique straight line segment connecting two points in $\mathbf{R}^N$.

While the commonly used metrics in Equations (2) and (3) are perfectly valid for use on a lattice (since a lattice is a subset of $\mathbf{R}^N$), they are not the best possible metrics for our purposes because: (1) we seek a metric that has a simple representation in general lattice coordinate systems; (2) we seek a metric that reflects the minimal distance a module is able to travel while staying within the lattice.

As an example of how δ_1 and δ_2 do not satisfy condition (1) above, consider what happens if the

unit vectors defining the skewed axes of a lattice coordinate system are expressed in an orthonormal basis for $\mathbf{R}^N$. The vectors form the columns of a matrix $\mathbf{V} = [\vec{v}_1, \ldots, \vec{v}_N] \in \mathbf{GL(N)}$ (the group of all invertible $N \times N$ matrices). In this case, the Euclidean metric is written as

$$\delta_2^s(a, b) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} V_{jk}V_{ji}(a_i - b_i)(a_k - b_k)}, \quad (4)$$

where $\{a_i\}$ and $\{b_i\}$ are the coordinates of the points a and b measured along the skewed lattice coordinate axes, V_{ij} are the elements of $\mathbf{V}$, and the superscript s indicates the skewed coordinate system. This superscript will be dropped when there is no ambiguity. Clearly, this computation is much more cumbersome than when dealing with orthogonal coordinate axes.

As an example of how the Euclidean metric does not satisfy condition (2) given above, consider the shortest path the center of a hexagonal module can make while staying within a hexagonal lattice. It will generally have to "zigzag" back and forth from one lattice point to another, instead of following a straight line. The δ_1 metric is even worse in terms of being cumbersome and not reflecting the least distance between lattice points, because it is intrinsically related to orthogonal coordinates. Thus, standard metrics are not particularly useful for this problem.

If we construct a *lattice connectivity graph*, i.e., a graph with vertices at lattice points, and edges that are straight lines connecting all neighboring vertices (of length given by the Euclidean metric), the lattice metric can be defined as the distance measured along *shortest paths* in this graph. More precisely, the *lattice metric function*, $\delta_L(a, b)$, is defined for a given lattice to be the length of the shortest path in the lattice connectivity graph that connects lattice points a and b. As is the case with module kinematics, this is best developed on a case by case basis. In section 4, lattice metrics are very important in establishing a lower bound on the number of module motions required to reconfigure a metamorphic system. We now illustrate the concept of a lattice metric with a hexagonal lattice in the plane.

Within the coordinate system shown in Figure 5(a), the Euclidean metric is

$$\delta_2^s(a, b) = ((\Delta i)^2 + (\Delta j)^2 - \Delta i \Delta j)^{1/2},$$

where $a = \vec{a} = (i_1, j_1)^T$, $b = \vec{b} = (i_2, j_2)^T$, $\Delta i = i_1 - i_2$, and $\Delta j = j_1 - j_2$.

The extra term added to the usual "$((\Delta i)^2 + (\Delta j)^2)^{1/2}$" (inside the square root) results from the fact that one of the coordinate axes is skewed so that there is not a 90° angle between the axes. The above formula for δ_2^s is calculated easily from trigonometry, or from Equation 4, where in this case $\vec{v}_1 = (\sqrt{3}/2, -1/2)^T$, and $\vec{v}_2 = (0, 1)^T$. The skewed axes are indicated by the solid lines in Figure 5(a). The taxicab norm is exceedingly complicated in this coordinate system. However, we would like to have an analogous system that will measure the true distance a module must travel within the regular lattice formed by all other modules if it is to roll from one lattice space to another.

This is achieved quite simply by taking the difference between the lattice coordinates of the two points and treating the difference as a vector centered at the origin of Figure 5(a). For the hexagonal lattice, a different representation of the distance function is used depending on which sextant this vector falls in, as stated below:

$$\delta_L(\vec{a}, \vec{b}) = |\Delta i| \text{ when } \vec{a} - \vec{b} \text{ in } 1, 4$$

$$\delta_L(\vec{a}, \vec{b}) = |\Delta j| \text{ when } \vec{a} - \vec{b} \text{ in } 2, 5.$$

$$\delta_L(\vec{a}, \vec{b}) = |\Delta i| + |\Delta j| \text{ when } \vec{a} - \vec{b} \text{ in } 3, 6.$$

It is conceptually trivial to show that these satisfy the definition of a metric given previously by direct calculation for each case. Furthermore, it is easy to see that this metric is bounded below by the Euclidean metric as demonstrated for sextant 1 (in which $-\Delta i \le \Delta j - \Delta i \le 0$ and $\Delta j \ge 0$):

$$\begin{aligned}\delta_2^2 &= |\Delta i|^2 + |\Delta j|^2 - \Delta i \Delta j \\ &= |\Delta i|^2 + \Delta j(\Delta j - \Delta i) \le |\Delta i|^2 = \delta_L^2.\end{aligned}$$

Furthermore, if we compare the minimal value of the Euclidean metric with the lattice metric in sextant 1 presented above for any fixed Δi, we see that

$$\frac{\partial \delta_2^2}{\partial(\Delta j)} = 2\Delta j - \Delta i = 0,$$

indicating that

$$\min_{\Delta j}(\delta_2^2) = \frac{3}{4}(\Delta i)^2 = \frac{3}{4}(\delta_L)^2.$$

Thus, it is clear that in sextant 1, the new metric is bounded below and above as follows:

$$\delta_2 \leq \delta_L \leq \frac{2}{\sqrt{3}}\delta_2.$$

Furthermore, it can be reasoned from symmetry that this holds in the other five sextants of the plane. Note that these bounds are much closer than those between Euclidean and Manhattan metrics in the plane ($\delta_2 \leq \delta_1 \leq \sqrt{2}\delta_2$).

As an example of the above relationship between Euclidean and Lattice metrics for the hexagonal case, consider the distance between the lattice points with coordinates (0, 0) and (2, 3). The Euclidean metric yields the distance $\sqrt{7}$, while the Lattice metric yields 3. Clearly the above inequalities hold: $\sqrt{7} < 3 < 2\sqrt{7/3}$. Note that the lattice distance will always be an integer. Also, there is generally not a unique minimal length path between arbitrary points in the lattice when using the lattice metric. For example, the line segments connecting the following coordinates are all "geodesics" in the lattice metric: {(0, 0), (0, 1), (1, 2), (2, 3)}, {(0, 0), (1, 1), (2, 2), (2, 3)}, {(0, 0), (1, 1), (1, 2), (2, 3)}.

3.2. Defining Distance Between Configurations

In the previous subsection, we defined the concept of a lattice metric—a measure of distance between modules in a lattice. In this subsection, we define a measure of distance between *configurations* of a metamorphic system. Each configuration of n modules is defined by the collection of n connected lattice spaces that it fills. That is, we do not distinguish between different modules, and any permutation of labels has no effect on configuration. Therefore, two configurations with the same shape and relative position in space are said to be the same. One metric on the set of all configurations, containing elements A and B, is given by

$$\delta_C(A, B) = M_{min}(A, B),$$

where M_{min} is defined as the fewest moves needed to reconfigure while observing locomotion constraints (which are described in detail in subsection 4.1).

Unfortunately, this metric has no representation other than explicitly solving a computationally explosive problem, and recording the number of moves which is minimal. If in fact we could do this in a reasonable amount of computational time, there would be no need for the formulations of section 4, and the optimal reconfiguration problem could simply be formulated as a shortest path problem on a graph. But this is not possible due to the computational complexity of this approach.

There are two reasons why this problem is computationally intractable. First, consider the problem of enumerating all possible connected configurations of n modules with fixed base. This problem is related to the *cell growth* problem in graph theory and combinatorics. The cell growth problem deals with finding the number of "animals" (configurations) that can be formed with n cells. To the best of our knowledge, this general problem is still an unsolved problem in combinatorics.[33] However, to get an idea of the complexity involved, let's consider the specific case of planar configurations in a hexagonal lattice. In Lunnon[34] an algorithm which is a modification of brute force enumeration is used to count the number of such configurations (referred to as *polyominoes* in some research[34,35]) that form an equivalence class under translation (*fixed polyominoes*). Harary and Palmer[33] tried to simplify the problem and came up with a recurrence relationship for configurations without *peri-connexions*, i.e., no three hexagons having a common point and no rings formed.

Even with these simplifying assumptions, the number of configurations grows dramatically with the number of modules. In Lunnon[34] the following results for the number of configurations, $C(n)$, forming an equivalence class under translation consisting of n modules are presented in Table I.

In the simplified case dealt with in Harary and Read,[36] the results are asymptotic to

$$\frac{(2n-1)!}{(n-1)!(n+1)!}\left(\frac{5}{4}\right)\sqrt{5}. \tag{5}$$

Table 1.
Growth of number of configurations with number of modules.

n	C(n)
1	1
2	3
3	11
4	44
5	186
6	814
7	3652
8	16689
9	77359
10	362671
11	1716033
12	8182213

n = number of modules.
$C(n)$ = number of configurations.

The above illustrations suggest a very rapid (nonpolynomial) growth in the number of different configurations as a function of the number of modules. However, establishing the *number* of different configurations consisting of n modules is just the first part of the problem. The second part of the problem is to represent configurations as nodes of a network, or vertices of a graph, i.e., each vertex representing a different configuration. In this graph, edges exist between two vertices if and only if the corresponding configurations are one move apart. That is, one module motion along the exterior of a configuration defines a new configuration, and the vertices corresponding to these two configurations are connected with an edge of the graph.

Thus, the minimal number of moves could be formulated as a shortest path problem on this graph (See Gibbons,[37] or any good book on graph theory). The problem with this approach is: (1) there does not currently exist an effective method for enumerating all possible connected configurations (which is even more explosive than enumerating all nonisomorphic configurations); and (2) even if (1) could be solved, the solution of the shortest path problem (while only being $O(m^2)$ where m is the number of vertices[38]) is computationally intractable in this case because the number of vertices grows exponentially in the number of modules.

Therefore, our only hope for solving this problem is to define and use heuristic algorithms. However, we would still like to make sure that these heuristics are in some sense "efficient," even though there is no practical way of even determining the actual fewest number of module motions required to reconfigure between desired configurations composed of large numbers of modules.

The next section establishes lower and upper bounds on the fewest number of moves, which will give us the ability to reject heuristic solutions that lie outside the upper bound, and will allow us to characterize how many moves are required for a heuristic in comparison with the lower bound.

4. GENERAL FORMULATION OF THE MOTION PLANNING/RECONFIGURATION PROBLEM

In this section we formalize the motion planning problem for metamorphic robotic systems using the kinematic/metric properties derived in the previous section. The formulation presented here is *not* constructive in the sense that it does not define any particular motion planning heuristic. Rather, it is an evaluation tool that can be used to rate the performance of any heuristic.

Subsection 4.1 states the constraints on module motion. These are, in a sense, discrete rolling constraints (as exemplified in Fig. 2). In subsections 4.2 and 4.3, respectively, we establish lower and upper bounds on the fewest number of moves required to reconfigure a general metamorphic system between arbitrary configurations. Section 5 will illustrate the general methodology presented here with examples.

4.1. Motion Constraints

The kinematic constraints governing the motion of one module over the surface of a collection of other modules are:

- Modules can only move into spaces that are not already occupied by other modules.
- Every module must remain connected to at least one other module, and at least one of the modules must stay connected to the fixed base from which the collection of modules originated.
- Only a single module may move one lattice space per timestep, and it achieves this motion by mating faces to faces (or in the planar case, edges to edges).

The first two constraints above are given by physical properties of the system. The third property is an artificial restriction imposed to make the problem tractable. It should also be noted that to date we have only demonstrated single module motions with hardware.

Under these constraints, the motion planning/self-reconfiguration problem becomes: *determination of the sequence of module motions from any given initial configuration to any given final configuration in a reasonable (preferably minimal) number of moves*. Factors that complicate this are (1) module motions do not commute; and (2) modules are very restricted in their movements, to the extent that composition of motions is not always possible, i.e., the motion of one module may prevent allowable motion of a neighboring module. Examples of these situations are shown in Figure 6. Therefore, the powerful tools of group theory and commutative algebras cannot be applied in any obvious way.

As was seen in the previous section, achievement of our motion planning/self-reconfiguration goal cannot be guaranteed in the "fewest" moves because the computational complexity is too great for large numbers of modules. Therefore, we desire to bound from below and above the minimal number

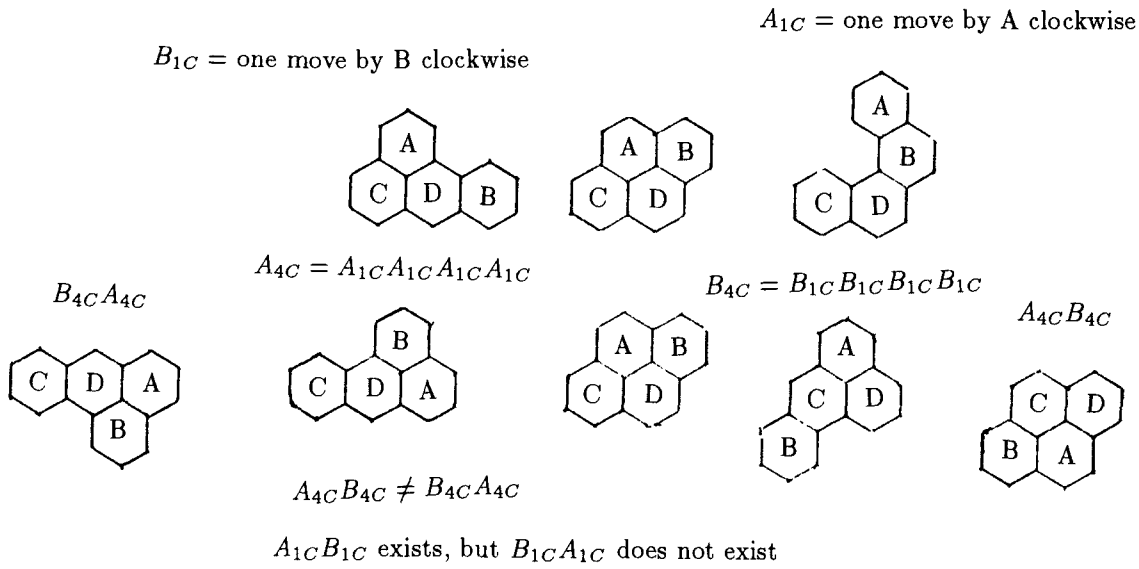


Figure 6. Examples of noncommutative module motions.

of moves required to get from any connected configuration to any other with the same number of modules. We also desire that these bounds have the following properties:

- They can be computed quickly.
- They are a function of geometric parameters of the initial and final configurations, e.g., perimeter, area, moments of area, intersections, unions, etc.
- They should couple our concepts of distance between modules/lattice points and distance between configurations.
- They should be as tight as possible.

The next two subsections establish bounds that satisfy the above conditions.

4.2. Lower Bounds on the Minimal Number of Module Motions

A good lower bound on the minimal number of moves required to reconfigure a metamorphic robot is obtained by using the lattice metric and concepts of optimal assignment. The lower bound presented here is based on the fact that the minimal number of moves required for a single module motion in a lattice will be no less than the lattice distance between the initial and final spaces. Furthermore, if it were possible to track the sequence of motions of an optimally reconfiguring metamorphic robot, we could compute the lattice distance between each module in its initial and final lattice spaces, and the sum of all these distances would be a lower bound on the total number of module motions. Since this is not possible, we will assign modules in two configurations in such a way that the sum of the lattice distances between matched modules is minimized over all possible matchings. This minimal sum will be at most the aforementioned lower bound. Since this *is* something that can be computed relatively efficiently (in at most $O(n^3)$ computations, see a previous work[39] for details), this is the lower bound we will use. The basic approach is summarized below.

Let the present configuration of the robot be described by the set of modules A, where $a_i \in A$ represents a module in the configuration for $i = 1, \ldots, n$, and $\vec{a}_i$ is its corresponding position vector. Let the new configuration be defined by the set B, where $\vec{b}_j \in B$ for $j = 1, \ldots, n$ represents a module in the new configuration, and $\vec{b}_j$ is its corresponding position vector. A lower bound on the total number of moves required to go from A to B (or vice versa) is given by an optimal assignment of each element a_i in A to an element b_j in B, $f: A \rightarrow B$, such that the sum of the distances (as defined by the lattice metric) for the assignment is minimized. This lower bound is written as:

$$L_1(A, B) = \min_{\pi \in \Pi_n} \sum_{i=1}^{n} \delta_L(a_i, b_{\pi(i)})$$

where π represents a permutation of module labels and Π_n is the set of all permutations of a set with n

elements. Equivalently, this can also be treated as finding a perfect matching in a weighted *bipartite graph* $G = (A, B)$,[37] such that the sum of the weights of the matching is minimized. In a previous work[39] we show that $L_1(A, B)$ and variants of it are metrics on the set of all configurations of modules that can be used together with optimization techniques to define heuristic reconfiguration strategies.

In cases where the overlap of the two configurations is on the same order as the number of modules, it is worth noting that $L_1(A, B)$ can be computed by matching overlapping modules in two configurations to each other. In effect this means the matching need only be performed between modules not in the overlap of two configurations, since the contribution to $L_1(A, B)$ from the overlapping parts is zero. The proof of this fact is given in a previous work.[39]

The complexity of optimal assignment algorithms is order $O(n^3)$ where n is the number of modules.[37] A simpler but less "tight" lower bound having complexity $O(n)$ is described below.

Recall that a lattice is a discrete subspace of the vector space $\mathbf{R}^N$. Furthermore, all of the metrics used in this article to define distance between lattice points can be extended to arbitrary points in $\mathbf{R}^N$ by evaluating the metric functions with vectors of real numbers instead of integers. These extended metrics are used to define norms on $\mathbf{R}^N$, and we can write for any given $\vec{x}, \vec{y} \in \mathbf{R}^N$, $\|\vec{x} - \vec{y}\| = \delta(\vec{x}, \vec{y})$. Useful properties that result from the above identification include:

$$\begin{aligned} \|\vec{x}\| &= \delta(\vec{x}, \vec{0}), \\ \delta(\vec{x}, \vec{y}) &= \|(\vec{x} - \vec{y}) - \vec{0}\| = \delta(\vec{x} - \vec{y}, \vec{0}), \\ \delta(c\vec{x}, c\vec{y}) &= \|c(\vec{x} - \vec{y})\| = c\|\vec{x} - \vec{y}\| = c\delta(\vec{x}, \vec{y}). \end{aligned} \quad (6)$$

where $c \in \mathbf{R}^+$, and $\vec{0}$ is the position vector of the base module. We will use these facts shortly.

Let us define the *centroid* of a collection of modules that define a configuration, A, to be

$$\vec{a}_c = \frac{1}{n}\sum_{i=1}^{n} \vec{a}_i,$$

where $\vec{a}_i$ is the position vector to the center of the module a_i of configuration A with respect to a coordinate system at the fixed base module. This coordinate system can be either a skewed lattice coordinate system, or an orthogonal one. In the former, we will denote vectors with the superscript L, and in the latter with superscript O. Thus, $\vec{a}_i^O = \mathbf{V}\vec{a}_i^L$, and because of linearity, $\vec{a}_c^O = \mathbf{V}\vec{a}_c^L$, where the elements of the matrix $\mathbf{V}$ were used in Equation (4). A nice property of the centroid is that it can be used to generate a lower bound on the quantity we seek. Namely, if we are given two configurations, $A = \{a_i\}$ and $B = \{b_i\}$, and if the modules are matched *in any arbitrary way*: $a_i \leftrightarrow b_i$ for $i = 1, \ldots, n$, we see that

$$\begin{aligned} \sum_{i=1}^{n} \delta(\vec{a}_i, \vec{b}_i) &= \sum_{i=1}^{n} \|\vec{a}_i - \vec{b}_i\| = \sum_{i=1}^{n} \delta(\vec{a}_i - \vec{b}_i, \vec{0}) \\ &\geq \delta\left(\sum_{i=1}^{n} (\vec{a}_i - \vec{b}_i), \vec{0}\right) = \delta\left(\sum_{i=1}^{n} \vec{a}_i, \sum_{i=1}^{n} \vec{b}_i\right) \\ &= \delta(n\vec{a}_c, n\vec{b}_c) = n\delta(\vec{a}_c, \vec{b}_c). \end{aligned}$$

The above inequality resulted from the triangle inequality and Equation 6. Thus, if we find the distance between centroids of each configuration, and multiply by the number of modules in each, this will be less than or equal to the sum of distances computed using any matching (including the minimal one). Therefore, the distance between centroids multiplied by the number of modules can be used as a lower bound on the minimal number of module motions required to reconfigure between A and B. This is written as

$$L_2(A, B) = \lfloor n\delta(a_c, b_c) \rfloor$$

where $\delta(a, b)$ can be either a lattice or Euclidean metric, and the greatest integer function $\lfloor \cdot \rfloor$ is used because the minimal number of moves must be an integer.*

This approach will yield good results when the two configurations are diametrically opposed, i.e., when the configurations are very different and the centers of mass of the two configurations are not close. However, it is possible for the centers of mass to be close and the configurations to be very different. In this case $L_2(\cdot, \cdot)$ will be a gross underestimate of the required number of moves. For instance, two serial configurations with an overlap at their center of mass, but rotated by an arbitrary nonzero angle, will need a very large number of moves $O(n^2)$ to reconfigure from one to the other. However, the centroid approximation will yield a lower bound of zero. On the other hand, if two serial configurations protrude from opposing sides of their overlap, then the centroid approximation to the lower bound will be very good, and it can be used to save some compu-

*In fact, $n\,\delta_L(\vec{a}_c^L, \vec{b}_c^L)$ will always be an integer, but $n\delta_2(\vec{a}_c^O, \vec{b}_c^O)$ will generally not be.

tational effort since it can be computed with $O(n)$ calculations.

4.3. Upper Bounds on the Minimal Number of Module Motions

In this subsection, we derive two closed-form upper bounds on the minimal number of module moves required to reconfigure between arbitrary configurations. These upper bounds are functions of the initial and final "perimeters" of the configurations, and the largest possible perimeter that a connected configuration of n modules can have. In addition, these bounds are functions of the number of modules in the largest simply connected overlap between the two configurations, I'_{AB}. We begin by formalizing some intuitive concepts.

Definition: The "exterior" of a configuration (collection of modules) is the union of all lattice spaces not within the configuration of modules, but at most one space distant from a module in the configuration.

Definition: A "partial perimeter" of a configuration is the number of moves required for an imaginary module to visit every lattice space in a connected component of the exterior while generating a closed path. The "perimeter," $P(n)$, of a configuration is the sum of all partial perimeters.

Definition: A "movable module" is a module that can move from its current lattice space to an adjacent space in the exterior of the current configuration without disconnecting itself or any other modules in the process.

Definition: A "maximal simply-connected overlap," of two configurations A and B is any simply connected (i.e., has no loops or voids) subset of modules contained in $A \cap B$ that has the greatest number of modules and contains the base module. (This set need not be unique, but the number of modules in this set, denoted I'_{AB}, is unique.)

In the case of planar modules, the above definition of perimeter reduces to simply the sum of the minimal number of empty spaces that surround the collection of modules, together with the number of empty spaces in the interior of each "hole" in contact with modules if the configuration has loops. For planar illustrations of each of the above definitions, see Figures 7 and 8.

We are now armed with the major definitions needed to derive an upper bound on the minimal number of moves required to reconfigure from any configuration to any other with the same number of modules.

There are three perimeters that will be of particular interest to us for a connected configuration consisting of n modules: (1) the perimeter of a connected subset of the initial configuration containing $n - 1$ modules (where one movable module has been removed such that the resulting perimeter is minimized): $P_i(n - 1)$; (2) the perimeter of a connected subset of the final configuration containing $n - 1$ modules (where one movable module has been removed such that the resulting perimeter is minimized): $P_f(n - 1)$; and (3) the maximal perimeter that a connected configuration of $n - 1$ modules can have: $P_{max}(n - 1)$. The Appendix proves that for systems

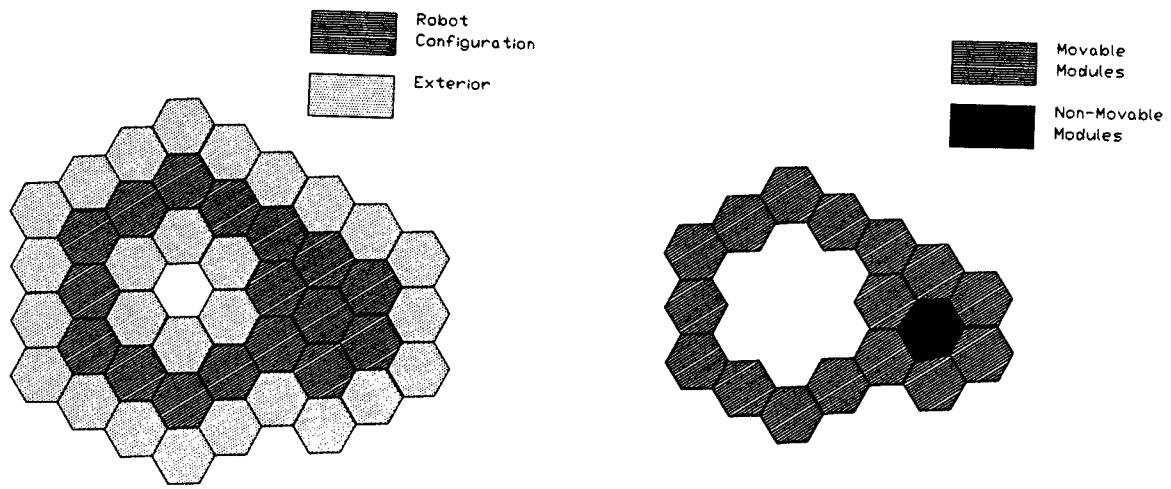


Figure 7. Exterior and movable modules.

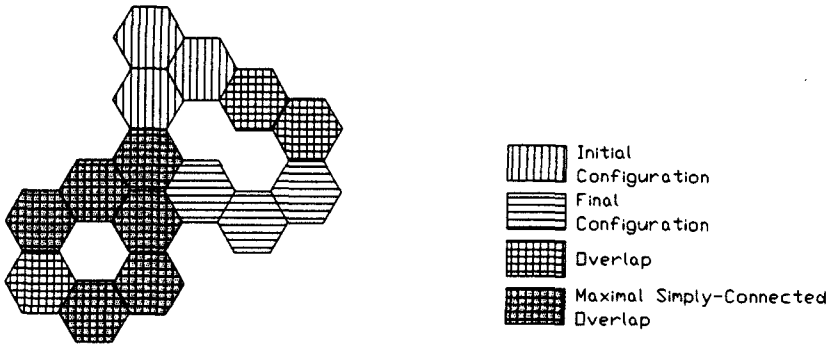


Figure 8. Maximal simple-connected overlap.

composed of planar hexagonal modules, $P_{max}(k) = 2(k + 2)$, and that this value occurs for serial configurations without branches or loops. For the sake of notational compactness, we will refer to the above functions without their arguments when there is no ambiguity.

Theorem: *An upper bound on the minimal number of moves, $M_{min}(A, B)$, required to reconfigure between any two configurations A and B with n modules and maximal simply connected overlap with I'_{AB} modules is: $U_1(A, B) = (n - I'_{AB})(P_{max}(n - 1))/2$.*

Proof: When a given module moves, it is not counted in the perimeter it must traverse. By definition, there is no longer a path that a module can take in the exterior of a configuration than the full perimeter. Thus, if a module takes a path to, and returns from, any arbitrary lattice space in the exterior of the configuration, it will take at most $P_{max}(n - 1)$ moves because it is traversing a configuration with $n - 1$ modules. Any minimal length path connecting two different points will thus be at most half of this length, because either the circuit has equal length on outgoing and return paths, or else we can always choose the smaller one. This process is repeated the fewest number of times needed to reconfigure (for the tightest bound). This number is the number of modules not in a maximal simply connected overlap of the two configurations (which is $n - I'_{AB}$). The other modules can simply remain fixed. ■

We impose the restriction that only the largest simply connected overlapping region including the base (as opposed to the whole overlap) need not move for the following reason. If the overlap is not simply connected, modules from one configuration might be inside a "hole," while modules of the other configuration could be on the outside. Similarly, if the overlap is not connected at all, there may be no way to reconfigure without moving the overlapping modules.

While $U_1(A, B)$ is a valid upper bound, it can be made tighter by incorporating information that is readily available, i.e., the initial and final perimeters. Let us define the maximal amount of change that the motion of one module can make to the perimeter of a configuration to be M. Furthermore, let us choose the initial and final perimeters traversable by a module to be the smallest of all possible perimeters of $n - 1$ connected modules contained in the initial and final configurations, respectively. We can do this without loss of generality by examining all modules that are able to move in the initial and final configurations, and choosing the first and last modules so that the perimeter created by excluding these modules is minimal. The following theorem incorporates all this information:

Theorem: *A tighter upper bound on the minimal number of moves required to reconfigure between any two configurations with n modules, maximal simply connected overlap consisting of I'_{AB} modules, and initial, final, and maximal possible perimeters P_i, P_f, and P_{max} is:*

$$U_2(A, B) = \min\{U'_2(A, B), U''_2(A, B)\},$$

where

$$\begin{aligned} U'_2(A, B) = [&P_i i_1 + M i_1(i_1 - 1)/2 + P_{max}(i_2 - i_1) \\ &+ (n - I'_{AB} - i_2)(P_f + nM - I'_{AB}M) \\ &- M(n - I'_{AB})(n - I'_{AB} - 1)/2 \\ &+ M i_2(i_2 - 1)/2]/2 \end{aligned} \tag{7}$$

and

$$U_2''(A, B) = [P_i i_3 + Mi_3(i_3 - 1)/2 + (P_f + nM - I'_{AB}M)(n - I'_{AB} - i_3) - M(n - I'_{AB} - 1)(n - I'_{AB})/2 + Mi_3(i_3 - 1)/2]/2 \quad (8)$$

where i_1 and i_2 are integers defined by the expressions:

$$P_i + i_1 M = P_{max}$$

$$P_{max} = P_f + (n - I'_{AB} - i_2)M$$

$$P_i + i_3 M = P_f + (n - I'_{AB} - i_3)M.$$

M is the largest amount by which the perimeter of a configuration can change *(increase or decrease) by the motion of one module.*

Proof: The half perimeter that the first module traverses is bounded from above by $p(0) = P_i/2$, the second will be bounded from above by $p(1) = (P_i + M)/2$, and the $j + 1^{st}$ will be bounded from above by $p(j) = (P_i + jM)/2$, until j is large enough that either $P_i + i_1 M = P_{max}$, or $P_i + i_3 M = P_f + (n - I'_{AB} - i_3)M$ for some integers $j = i_1$ or $j = i_3$. That is, until the perimeter reaches its maximal possible value, or it reaches such a value that the perimeter must start to decrease to attain the perimeter of the final configuration with moves of the remaining modules. The above conditions have the geometric meaning of the position where the lines defined by $p(j) = P_i + jM$, $p(j) = P_{max}$, and $p(j) = P_f + (n - I'_{AB} - j)M$ intersect in the plane whose independent coordinate is j and dependent coordinate is p. In other words, the lines with slope $\pm M$ will intersect each other either above or below the line with zero slope, depending on the initial and final perimeters (the intercepts of the lines) and M (the slope of the lines). Since the moves associated with each half perimeter are added to the total, and we seek a bound on the minimal total moves, we seek the planar figure that will have the least area bounded by these three lines and the j axis. This will either be a triangle with peak below P_{max}, or a trapezoid with P_{max} as the top line. The expressions in the statement of the theorem correspond to these cases, and are derived below, and the two cases are depicted in Figure 9.

(Case 1) Triangle: Suppose that the lines given by the equations $p(j) = P_i + jM$ and $p(j) = P_j + (n - I'_{AB} - j)M$ (which are upper bounds on the perimeter) intersect each other below the line $p(j) = P_{max}$. By denoting the value of j where these lines intersect as i_3, the perimeters will be summed using the first line until i_3, and then we will switch to the second line. The result of this summation is:

$$\sum_{j=0}^{n-I'_{AB}-1} p(j) = \sum_{j=0}^{i_3-1} (P_i + jM) + \sum_{j=i_3}^{n-I'_{AB}-1} (P_f + (n - I'_{AB} - j)M).$$

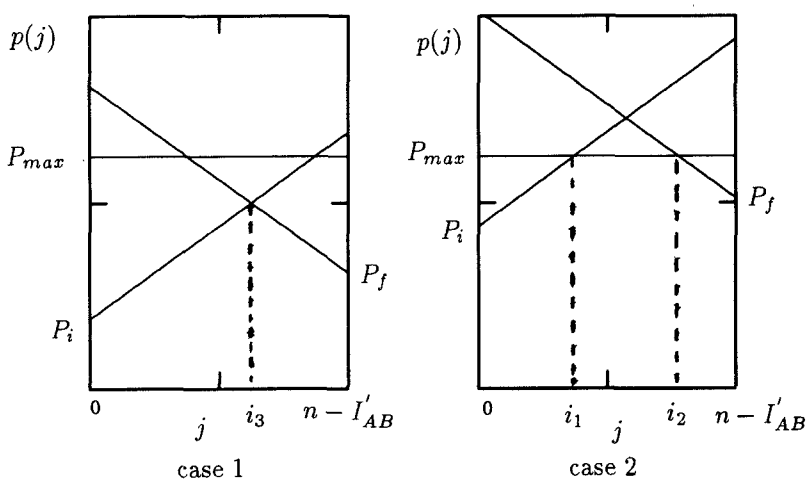


Figure 9. Graphical derivation of upper bounds.

These summations simplify using the formulas: $\sum_{j=k}^{r} 1 = r - k + 1$, $\sum_{j=k}^{r} j = r(r+1)/2 - k(k-1)/2$.

(Case 2) Trapezoid: By definition, i_1 is where the line with slope $+M$ intersects the horizontal line at P_{max}. Starting at $j = i_1$, the half perimeter traversed will be at most $P_{max}/2$ (since this is the maximal value possible). This is true until $P_{max} = P_f + M(n - I'_{AB} - i_2)$, for some integer $j = i_2$. This is where the line with slope $-M$ intersects the horizontal line at P_{max}. From $j = i_2$ on, the perimeter must decrease in the steepest way possible to be able to reach P_f using the unmoved modules. The perimeter for the remaining moves will be bounded from above by $p(j) = P_f + M(n - I'_{Ab} - j)$ for it to be possible to attain the final perimeter. If we sum up all three contributions from $j = 0$ to $n - I'_{AB} - 1$, Case 2 is proved. In other words, the upper bound is:

$$\sum_{j=0}^{n-I'_{AB}-1} p(j) = \sum_{j=0}^{i_1-1} (P_i + jM) + \sum_{j=i_1}^{i_2-1} P_{max} + \sum_{j=i_2}^{n-I'_{AB}-1} (P_f + (n - I'_{AB} - j)M),$$

which is simplified using the same formulas as case 1. ∎

The benefit of the two upper bounds discussed so far in this section is that they can be computed in $O(1)$ calculations, and they tell us right away that there will be at most $O(nP_{max})$ moves required to reconfigure *no matter what module design is chosen*. In the case of planar modules this will be $O(n^2)$. In addition to these easily computed bounds, we will often want to have *constructive* upper bounds on the minimal number of moves. That is, instead of conservatively assuming that half of the largest perimeter is traversed each time a single module moves to fill a space in the new configuration, we can construct intermediate configurations by having modules move along the perimeter (clockwise or counterclockwise) until they stop at a suitable place in the desired configuration. This will, by definition, require a fewer number of moves than the nonconstructive upper bounds presented above, but in the worst case will run to $O(n^2)$ calculations in the planar case.

While a number of constructive algorithms can be imagined, the simplest (although not necessarily most efficient) one for simply connected initial and final configurations is for a module to stop at the first space reached in the final configuration. The upper bound generated by counting the number of moves in this case is denoted $U_c(A, B)$. This will suffice for the purposes of illustration in the following sections.

5. APPLICATION TO MOTION PLANNING

This section introduces an application of the lower and upper bounds developed earlier in this article. Namely, the improvement, or *refinement*, of an existing sub-optimal move sequence is considered. As a general rule, heuristics can generate a sequence of moves that reconfigure a metamorphic system from one configuration to another. The application of the bounds presented in this article is to detect and improve inefficient spots in a heurustic move sequence. To formulate a refinement procedure based on lower and upper bounds on the number of required module motions, a few defintions are needed.

Suppose we are given an initial configuration I, and final configuration F. A *reconfiguration sequence* is defined as an ordered set of connected configurations $\{A_0, \ldots, A_{moves}\}$ for which (i) $\delta_C(A_0, I) = 0$, $\delta_C(A_{moves}, F) = 0$, and (ii) $\delta_C(A_i, A_{i+1}) = 1$ for $i \in [0, \ldots, moves - 1]$. where $\delta_C(\cdot, \cdot)$ is any configuration metric. A *reconfiguration subsequence* is a subset of a reconfiguration sequence: $\{A_i, \ldots, A_k\}$ for which $i \geq 0$ and $k \leq moves$, and in which all elements between A_i and A_k in the reconfiguration sequence are contained in the subsequence in their original order. A reconfiguration subsequence will be called *proper* if the above inequalities are strict inequalities.

Using these definitions, a *refinement* of a given reconfiguration sequence with initial and final configurations I and F that contains $moves + 1$ configurations is a reconfiguration sequence for which the first and last elements are I and F and the number of elements is $moves' + 1$ where $moves > moves'$.

Using the methods developed in this article, there are two easy ways to refine a given move sequence, $\{A_0, \ldots, A_{moves}\}$. These methods are referred to as *contraction* and *filtering*.

Contraction is defined as follows: For each $i \in [0, moves - 1]$, check whether or not $\delta_C(A_i, A_j) = 0$ for all $j > i$. If one or more subsequences of the form $\{A_i, \ldots, A_j\}$ exist such that $\delta_C(A_i, A_j) = 0$, then remove each subsequence $\{A_{i+1}, \ldots, A_j\}$, and renumber the configurations numbered $j + 1$ to $moves$ in the original sequence with the numbering $i + 1$ to $moves + i - j$ in the contracted sequence. This procedure is $O(n \cdot moves^2)$ because the intersection of

two configurations can be computed in $O(n)$, and there are $O(moves^2)$ comparisons. This procedure is referred to as a contraction because it takes one or more subsequences and reduces it to a single configuration, A_i, thus reducing (contracting) the size of the original sequence.

Filtering is defined as follows: Assume that a reconfiguration sequence is given. If $moves(I, F) > U_c(I, F)$, then we know the reconfiguration sequence can be improved since any constructive algorithm that rolls modules at most half way around the exterior of the configuration can be used to generate an upper bound. These bounds are then applied recursively for smaller and smaller reconfiguration subsequences until subsequences containing only two configurations are generated. If a subsequence is found for which this inequality does not hold, either: (i) run the heuristic a number of times until a subsequence can be generated for which the inequality holds (in particular, if the heuristic has stochastic properties, such as simulated annealing,[39] an indeterminate number of trial runs are needed), or (ii) substitute a subsequence generated using a constructive upper bound, such as the one described at the end of the previous section. Note that the tighter the upper bounds are, the "finer" the filter will be.

Using the contraction and filtering techniques above, reconfiguration sequences that are efficient enough to fall within the upper bounds are refined by "filtering out" those subsequences that detract from overall efficiency. While there is no guarantee to what extent these algorithms will improve a given reconfiguration sequence, they certainly do not detract performance, and run fast enough that they are not a computational burden. The next section considers examples.

6. EXAMPLES

In this section, a planar metamorphic system with hexagonal modules is used to demonstrate the methods developed in the previous sections. It is then demonstrated how the bounds established previously in this article are useful for: (1) evaluating the performance of heuristic algorithms, (2) "weeding out" inefficient reconfiguration procedures, and (3) improving the performance of a given reconfiguration procedure.

6.1. An Example of Lower and Upper Bounds

As an illustration of the lower bounds on the minimal number of moves, consider the following example. Figure 10(a) shows the initial configuration, Figure 10(b) shows the final configuration, and Figure 10(c) shows an arbitrary labeling of the modules in the two configurations. If we choose all possible matchings of the labels $\{1, 2, 3, 4\}$ with $\{1', 2', 3', 4'\}$ (of which there are 4!), we choose one for which the sum of the lattice distances between matched modules is minimized. The reader is encouraged to verify this by trying all possibilities. The optimal assignment in this case results from matching like numbered modules in Figure 10(c), i.e., $i \rightarrow i'$, and summing the lattice distances between all of them. In this case it is easy to see that $L_1(A, B) = 8$. It should be noted that enumerating all possible matchings is very inefficient ($O(n!)$). $O(n^3)$ algorithms for optimal assignment are used in practice.[39] If we observe the positions of each module in the lattice coordinate system attached to the base module (which is labeled as module 1) the centroid vectors for the two configurations are respectively $\vec{a}_c = \frac{1}{4}(-5, 1)$ and $\vec{b}_c = \frac{1}{4}(2, 0)$. Using the lattice metric we find that $L_2(A, B) = 4\delta_L(\vec{a}_c, \vec{b}_c) = \delta_L(4\vec{a}_c, 4\vec{b}_c) = 8$, which is the same as the value calculated using $L_1(A, B)$.

Now let us consider the upper bounds on the minimal number of moves computed for this configuration. In this particular case, $P_f(3) = P_i(3) = P_{max}(3) = 10$, and so no savings is gained by the second closed-form upper bound presented in section 4. The closed-form upper bounds are computed simply as: $U_1(A, B) = U_2(A, B) = 3 \cdot 5 = 15$. Construc-

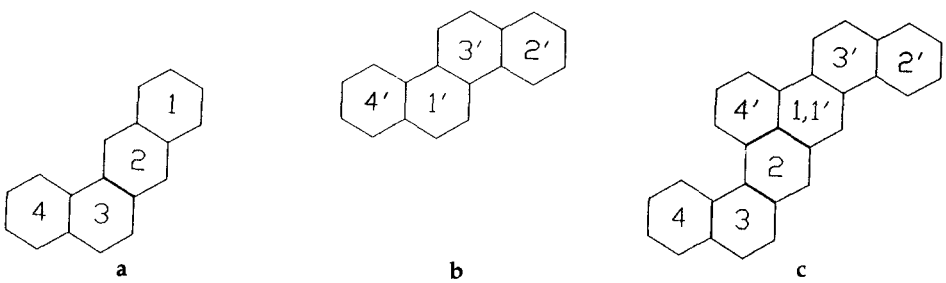


Figure 10. (a) Present configuration; (b) new configuration; (c) module labeling.

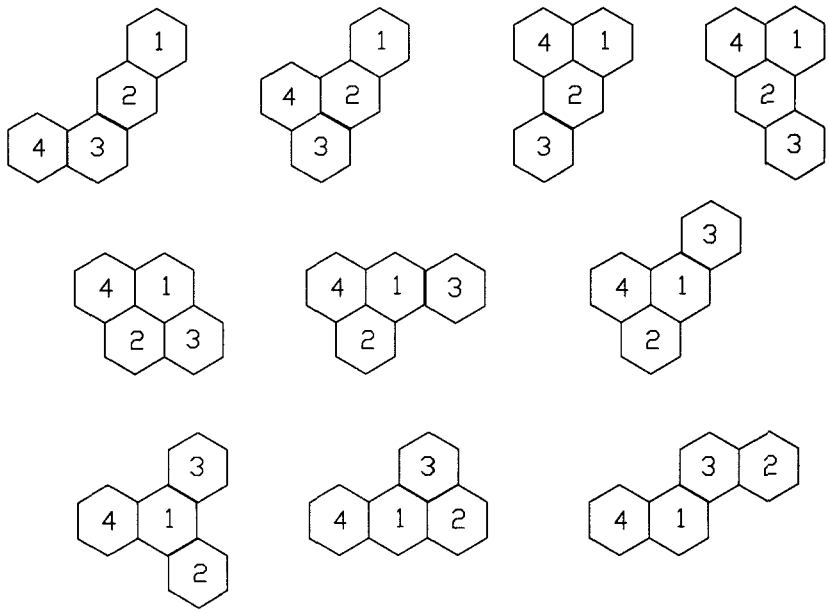


Figure 11. An optimal reconfiguration sequence.

tively computing an upper bound by rolling 4 to 4′, 3 to 3′, and 2 to 2′, one gets 9, thus, $8 \leq M_{min} \leq 9 < 15$ for this example. In fact, for such a small number of modules, it is easy to test all possible combinations of moves by hand, and one finds that $M_{min} = 9$. Figure 11 explicitly represents one possible strategy for optimal reconfiguration. The initial labels have been retained so that motions are easy to track.

6.2. Using Bounds to Improve Performance

As an example of the refinement procedure presented in section 5, consider the reconfiguration sequence presented in Figure 12. The lower bound is given by $L_1(I, F) = 4$. The closed-form upper bounds both give 8. Thus, this sequence is within bounds since $4 \leq 8 \leq 8$, and is not rejected. Although the reader is no doubt aware by looking at the initial

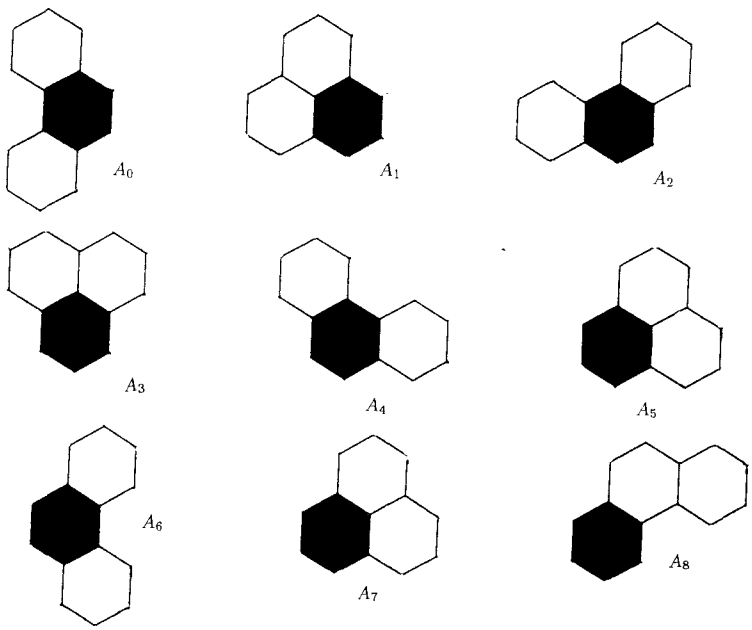


Figure 12. A suboptimal reconfiguration sequence.

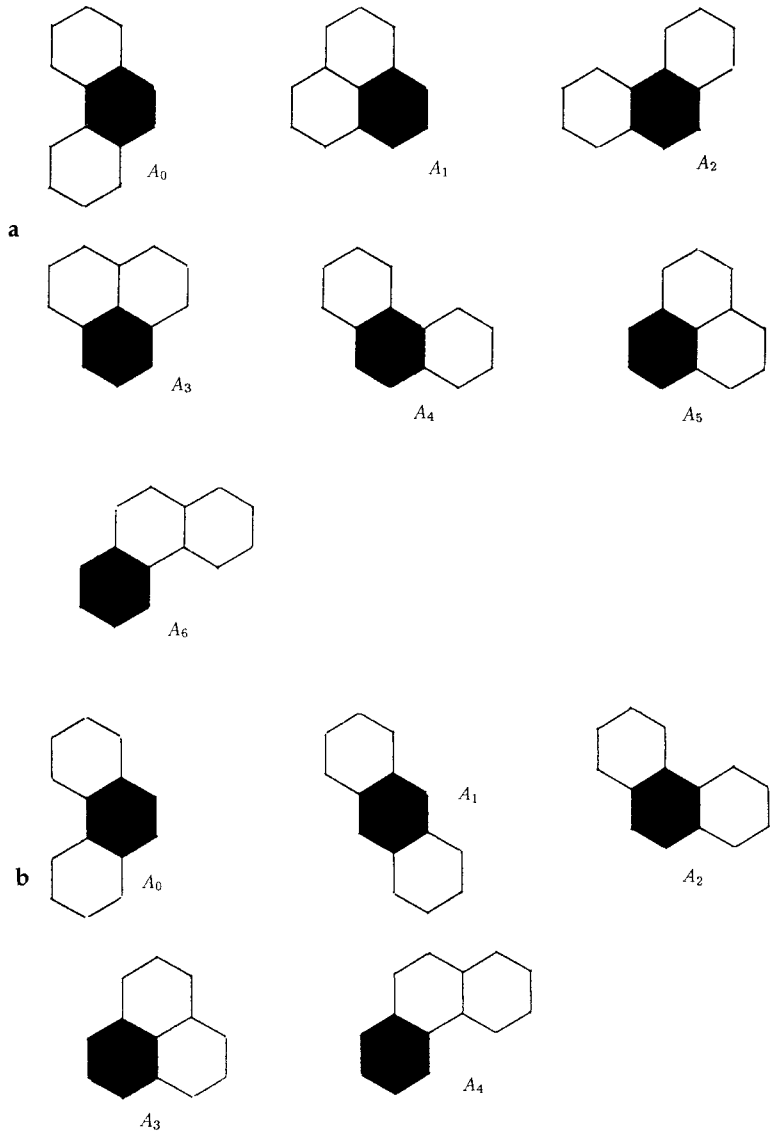


Figure 13. (a) A contracted sequence; (b) a filtered sequence.

and final configurations that an optimal sequence can be generated using fewer moves, we ignore this fact for the purpose of illustration.

Using the contraction algorithm, the resulting sequence is shown in Figure 13(a), where the subsequence $\{A_5, A_6, A_7\}$ was contracted from the original sequence, because it formed a loop in the configuration space of the metamorphic robot. The result is an improved sequence with 6 moves. Using the filtering algorithm we find that the whole sequence $\{A_1, A_2, A_3, A_4\}$ can be replaced with a two move sequence as shown in Figure 13(b) by using the constructive upper bounds outlined at the end of subsection 4.3. This is clear because $L_1(A_1, A_4) = 2 = U_c(A_1, A_4)$, and since lower and upper bounds are equal, this must be the minimal number of moves. Of course, the filtering algorithm could have been run first, but this would have precluded the illustration of contraction for this example.

7. CONCLUSIONS

The concept and kinematics of a metamorphic robotic system were developed in this article. Bounds on the fewest moves required to reconfigure from one configuration to another were established. These bounds are important because explicit solution for

the minimal number of moves becomes computationally infeasible when the number of modules is more than 15–20. It was illustrated how these bounds can be used to "weed-out" and improve inefficient motion planning strategies. Furthermore, the concepts developed here provide a framework from which efficient heuristics can be constructed.

Much work still remains in the development of motion planning/reconfiguration algorithms for metamorphic systems, and challenging issues remain in terms of mechatronic design and hardware implementation.

This work was made possible by National Young Investigator Award IRI-9357738 from the Robotics and Machine Intelligence Program at the National Science Foundation, and a Presidential Faculty Fellows Award.

APPENDIX: MAXIMUM PERIMETER OF METAMORPHIC ROBOT CONFIGURATIONS

Theorem: For a given number of modules, n, *a serial structure has the maximum perimeter.*

Proof: By induction on the number of modules.

Basis: For $n = 1$ and $n = 2$ only serial configurations can be formed. For $n = 3$, three distinct nonisomorphic configurations are possible, as shown in Figure A-1, along with their perimeters. The serial structure has the maximum perimeter among them.

Induction Hypothesis (IHOP): Let's assume that among all configurations with n modules, the serial structure has the maximum perimeter.

Consider a configuration of n modules with a perimeter P. Add one more module to this configuration. There are two distinct possibilities:

Case 1: The addition of the new module does not form any new loops. There are six possibilities for the position of the new module in the simply connected case, as show in Figure A-2.

(i) When the new module has only one neighbor, the perimeter increases by 2 as shown in Figure A-3(a).
(ii) When the new module has two neighbors, the perimeter increases by 1 as shown in Figure A-3(b).
(iii) When the new module has three neighbors, the perimeter remains the same as shown in Figure A-3(c).
(iv) When the new module has four neighbors, the perimeter decreases by 1 as shown in Figure A-3(d).
(v) When the new module has five neighbors, the perimeter decreases by 2 as shown in Figure A-3(e).
(vi) When the new module has six neighbors, the perimeter does not change as shown in Figure A-3(f).

The maximum change in perimeter, ΔP, is 2 when the new module is connected to only one neighbor. By IHOP, a serial structure with n modules has the maximum perimeter, say, P_s. Add one more module to form a serial structure with $n + 1$ modules. The new module can only be added at the ends and has one neighbor, increasing the perimeter by 2, i.e.

$$P_s + 2 \geq P + \Delta P \tag{1}$$

Case 2: The addition of the new module leads to the formation of a new loop in the configuration. The perimeter is then defined as the sum of the partial perimeters. Three distinct possibilities exist (other cases are not possible).

(i) When the new module has two neighbors, the perimeter decreases by 2 as shown in Figure A-4(a).
(ii) When the new module has three or four neighbors and only one loop is formed in the new configuration, the perimeter decreases by 3 and 4, respectively, as shown in Figure A-4(b).

Figure A-1. Three distinct configurations for $n = 3$.

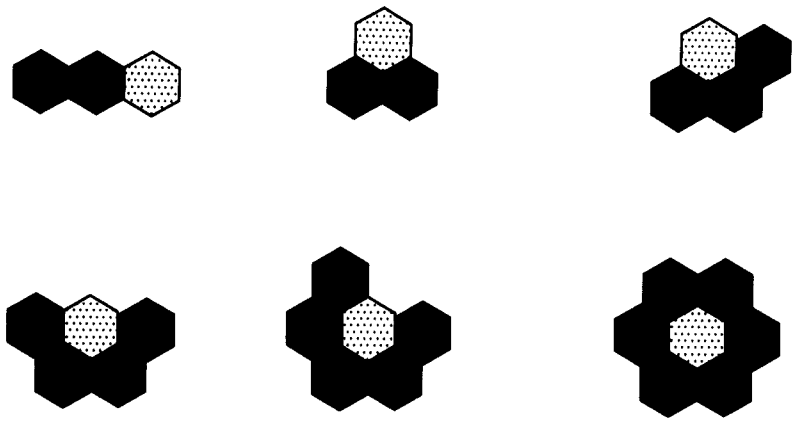
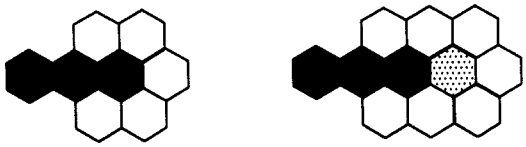
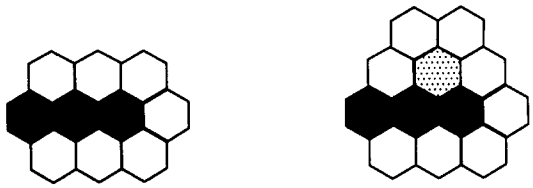


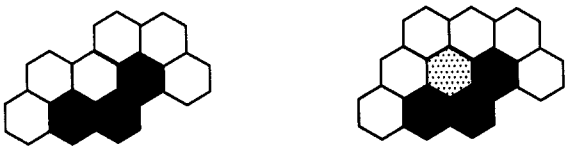
Figure A-2. The different neighbors possible for the new module.



(a) Change in perimeter with one neighbor.

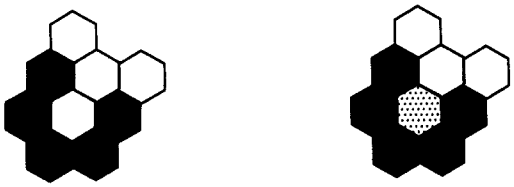


(b) Change in perimeter with two neighbors.



(c) Change in perimeter with three neighbors.

(d) Change in perimeter with four neighbors.



(e) Change in perimeter with five neighbors.

(f) Change in perimeter with six neighbors.

Figure A-3. Change in perimeter with different number of neighbors. The modules in black represent part of the pre-existing configuration, the hatched module is the new module added, and the white spaces represent part of the perimeter in the lattice.

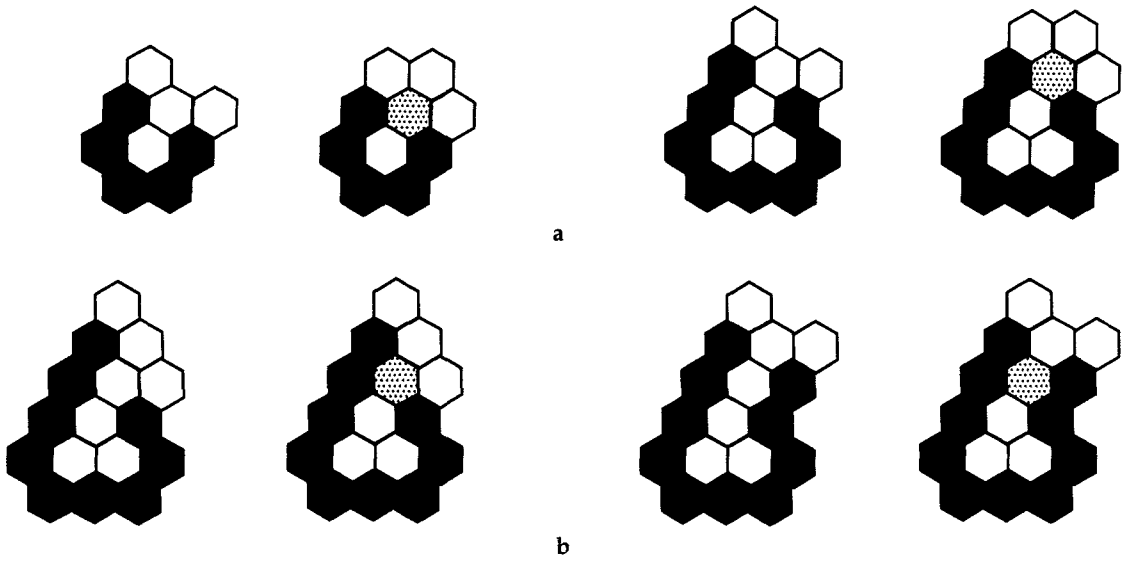


Figure A-4. Change in perimeter with two, three, and four neighbors. The modules in black represent part of the pre-existing configuration, the hatched module is the new module added, and the white spaces represent part of the perimeter in the lattice. (a) Change in perimeter ($\Delta P = 1, 2$ respectively) with two neighbors. (b) Change in perimeter with 3 and 4 modules, respectively.

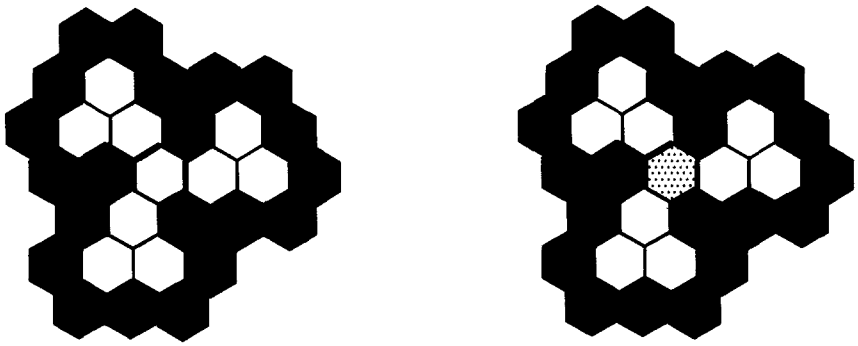


Figure A-5. Change in perimeter with formation of 3 loops.

It is easily observed that when the addition of the new module creates more than one loop, no increase in perimeter takes place, as shown by one example in Figure A-5.

Hence, no increase in perimeter takes place by the addition of a new module in case 2, i.e.

$$\Delta P \leq 0 \tag{2}$$

Hence from (1) and (2), for a given n, a serial structure has the maximum perimeter. ■

REFERENCES

1. S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," *Proc. IEEE Int. Conf. Rob. Autom.*, San Diego, CA, 1994, pp. 441–448.
2. M. Yim, "A reconfigurable modular robot with many modes of locomotion," *Proc. JSME Int. Conf. Adv. Mechatronics*, Tokyo, Japan, 1993, pp. 283–288.
3. M. Yim, "New locomotion gaits," *Proc. IEEE Int. Conf. Rob. Autom.*, San Diego, CA, 1994 (also see video proceedings).
4. A. Pamecha, Y. Kohaya, and G. S. Chirikjian, "Design of error tolerant coupling mechanisms for metamorphic robots," *IASTED Conf.*, Cancun, Mexico, 1995.

5. A. Pamecha and G. S. Chirikjian, "A metamorphic robotic system: Simulation and connector mechanism demonstration," *Proc. IEEE Int. Conf. Rob. Autom. (Video)*, Nagoya, Japan, 1995.
6. G. S. Chirikjian, "Kinematics of a metamorphic robotic system," *Proc. IEEE Int. Conf. Rob. Autom.* San Diego, CA, 1994, pp. 449–455.
7. B. Benhabib, G. Zak, and M. G. Lipton, "A generalized kinematic modeling method for modular robots," *J. Rob. Syst.*, **6**(5), 545–571, 1989.
8. R. Cohen, et al. "Conceptual design of a modular robot," *J. Mech. Des.*, March, 117–125, 1992.
9. L. Kelmar and P. K. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," *Proc. IEEE Int. Conf. Rob. Autom.*, 1988, pp. 663–668.
10. M. Sciaky, "Modular robots implementation," *Handbook of Industrial Robotics*, S. Nof, ed., John Wiley, New York, 1985, pp. 759–774.
11. K. H. Wurst, "The conception and construction of a modular robot system," *Proc. Int. Symp. Ind. Rob.*, 1986, pp. 37–44.
12. G. Beni, "Concept of cellular robotic systems," *Proc. IEEE Int. Symp. Intel. Control*, Arlington, VA, August 1988.
13. G. Beni and J. Wang, "Theoretical problems for the realization of distributed robotic systems," *Proc. IEEE Conf. Rob. Autom.*, 1991, pp. 1914–1920.
14. T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic systems," *Proc. IEEE Int. Conf. Rob. Autom.*, 1988, pp. 1581–1586.
15. T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," *Proc. IEEE Int Conf. Rob. Autom.*, 1990, pp. 662–667.
16. T. Fukuda, Y. Kawauchi, and F. Hara, "Dynamic distributed knowledge system in self-organizing robotic systems; CEBOT," *Proc. IEEE Int. Conf. Rob. Autom.*, 1991, pp. 1908–1913.
17. S. Hackwood and J. Wang, "The engineering of cellular robotic systems," *Proc. IEEE Int. Symp. Intel. Control*, Arlington, VA, 1988.
18. S. Hackwood and G. Beni, "Self-organization of sensors for swarm intelligence," *Proc. IEEE Int. Conf. Rob. Autom.*, 1992, pp. 819–829.
19. S. Hackwood and G. Beni, "Self-organizing sensors by deterministic annealing," *Proc. IROS*, Osaka, Japan, pp. 1177–1183.
20. H. Asama, et al. "Collision avoidance among multiple mobile robots based on rules and communication," *Proc. IROS'91*, 1991, pp. 1215–1220.
21. I.-M. Chen and J. W. Burdick, "Enumerating nonisomorphic assembly configurations of a modular robotic system," *Proc. IEEE/RSJ Int. Conf. Int. Rob. Syst.* Yokohama, Japan, 1993, pp. 1985–1992.
22. I-M. Chen, "Theory and applications of modular reconfigurable robots," Ph.D. dissertation, Caltech, Pasadena, CA, 1994.
23. K. W. Jeon, *The Biology of Amoeba*, Academic Press, New York, 1973.
24. A. A. Schaeffer, *Ameboid Movement*, Princeton University Press, 1920.
25. J. T. Bonner, *The Cellular Slime Molds*, Princeton University Press, 1967.
26. G. S. Chirikjian, "Theory and applications of hyperredundant robotic manipulators," Ph.D. dissertation, Caltech, Pasadena, Ca, 1992.
27. H. Gericke, *Lattice Theory*, Frederick Ungar Publishing, New York, 1966.
28. W. Miller, *Symmetry Groups and Their Applications*, Academic Press, New York, 1972.
29. M. B. Boisen and G. V. Gibbs, *Mathematical Crystallography*, Mineralogical Society of America, PI, Blacksburg, VA 1985.
30. J. E. Marsden, *Elementary Classical Analysis*, W. H. Freeman, New York, 1974.
31. E. F. Krause, *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*, Dover, New York, 1975.
32. J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.
33. F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, New York, 1973.
34. W. F. Lunnon, "Counting hexagonal and triangular polyominoes," *Graph Theory and Computing*, R. C. Read, ed., Academic Press, New York, 1972, pp. 87–108.
35. S. W. Golomb, *Polyominoes*, Scribner, New York, 1965.
36. F. Harary and R. C. Read, "The enumeration of tree-like polyhexes," *Proc. Edinborough Math. Soc.*, **17**, 1–13, 1970.
37. A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, New York, 1985.
38. E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms, Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.
39. A. Pamecha, I. Ebert-Uphoff, and G. S. Chirikjian, "A useful metric for modular robot motion planning, JHU Technical Report RMS-8-94-2.