Useful Metrics for Modular Robot Motion Planning

Amit Pamecha, *Student Member, IEEE*, Imme Ebert-Uphoff, and Gregory S. Chirikjian, *Member, IEEE*

***Abstract*— In this paper the problem of dynamic self-reconfiguration of a class of modular robotic systems referred to as *metamorphic* systems is examined. A metamorphic robotic system is a collection of mechatronic modules, each of which has the ability to connect, disconnect, and climb over adjacent modules. We examine the near-optimal reconfiguration of a metamorphic robot from an arbitrary initial configuration to a desired final configuration. Concepts of distance between metamorphic robot configurations are defined, and shown to satisfy the formal properties of a metric. These metrics, called configuration metrics, are then applied to the automatic self-reconfiguration of metamorphic systems in the case when one module is allowed to move at a time. There is no simple method for computing the optimal sequence of moves required to reconfigure. As a result, heuristics which can give a near optimal solution must be used. We use the technique of Simulated Annealing to drive the reconfiguration process with configuration metrics as cost functions. The relative performance of simulated annealing with different cost functions is compared and the usefulness of the metrics developed in this paper is demonstrated.**

***Index Terms*—Metrics, optimal assignment, self-reconfigurable robots, simulated annealing.**

I. Introduction

A *metamorphic* robotic system [7] 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 robot 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.

Metamorphic systems can be viewed as a large swarm (or colony) of connected robots which 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 communication functionality. This allows uniform treatment of modules in the planning problem.
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 which can carry load. This causes the collection of modules to act as a single physical object.

One of the module designs which satisfies all the above properties in the planar case consists of six links of equal length forming a six bar linkage, as shown in Fig. 1(a). Because of the hexagonal shape, the modules completely fill the plane without any gaps. As can be seen in Fig. 1(b), each module possesses three degrees of freedom which are controlled by placing actuators at alternate joints. This enables each module to move around another while remaining connected at all times during this motion. The modules are provided with electromechanical connectors actuated by D.C. motors. Each module carries male and female connectors on alternate links. Because of the symmetry of the module, male connectors always meet female connectors and vice versa [7]. In this particular implementation each male connector (T-shaped protrusion) is spring loaded to allow for alignment errors and to provide passive compliance during the reconfiguration sequence. For a hardware demonstration of the above design, see [27] and [28].

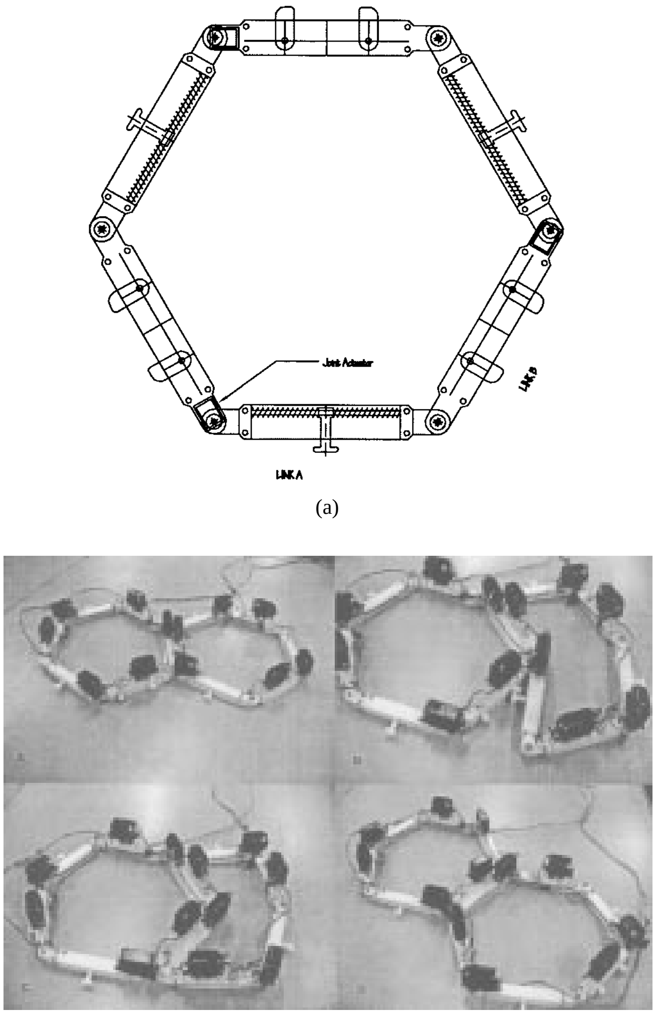
Potential applications of metamorphic systems composed of a large number of modules include: 1) obstacle avoidance in highly constrained and unstructured environments; 2) “growing” structures composed of modules to form bridges, buttresses, and other civil structures in times of emergency; 3) envelopment of objects, such as recovering satellites from space; and 4) Performing inspections in constrained environments such as nuclear reactors. Some of these applications are shown in Fig. 2.

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 [2], [9], [19], [31], [32]; 2) cellular robotic systems in which a heterogeneous collection of independent specialized modules are coordinated [3], [4], [12]–[14], [16]; and 3) swarm intelligence in which there are

Manuscript received October 27, 1995. This work was supported by a National Young Investigator Award IRI-9 357 738 and a 1994 Presidential Faculty Fellow Award through the Robotics and Machine Intelligence Program at the National Science Foundation. This paper was recommended for publication by Associate Editor V. Hayward and Editor S. Salcudean upon evaluation of the reviewers’ comments.

The authors are with Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218 USA.

Publisher Item Identifier S 1042-296X(97)03814-7.



(a)

(b)

Fig. 1. (a) An example design of planar module. (b) Hardware demonstration of motion involving two planar hexagonal modules.

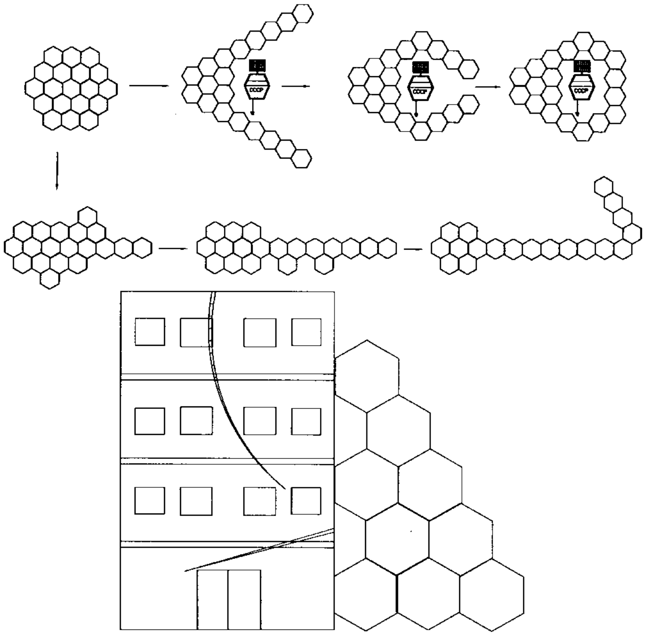


Fig. 2. Examples of metamorphic robot applications.

generally no physical connections between modules [18], [17], [1]. Most recently, two other types of modular reconfigurable robotic systems have been considered. Yim [33], [34] considered modular robots composed of a few basic elements which can be composed into complex systems, and used for various modes of locomotion. Murata *et al.* [25] considered a "fractal" system composed of modules with zero kinematic mobility, but which can "walk" over each other in discrete quanta due to changes in the polarity of magnetic fields. Chen and Burdick [6] provide a valuable tool for defining equivalence classes of modular robot configurations with the same shape and morphological function.

The concept of a metamorphic system differs from concepts in the 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 paper are applicable to other types of self-reconfigurable systems. For instance, the "fractal" modules introduced in [25], [26] exhibit all but the mobility requirement, and thus many of the methods in this paper apply.

This paper addresses issues in the 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 II, we review kinematic and motion planning issues pertaining to metamorphic robots and describe the complexity of the motion planning problem for metamorphic robots. In Section III, we define concepts of distance between configurations and discuss three types of configuration metrics. In particular, the "optimal assignment metric" is discussed and a method for evaluating it is illustrated with examples. In Section IV, we provide a formal proof that the concept of distance between configurations using optimal assignment is a metric on the set of all possible configurations. Section V describes some modified metrics useful for motion planning. Section VI discusses the application of the method of simulated annealing to the minimization of the cost function based on the metrics described in the earlier sections. This involves driving the distance between the current and the final configuration to zero. Section VII describes the results obtained from the implementation of a simulated annealing algorithm for three different configurations.

II. Problem Formulation and Mathematical Background

In this section, we formulate the general problem of describing metamorphic robot configurations, and characterize constraints on module motion. In Section II-A, we focus on the description of a given configuration. Section II-B examines constraints on module motion and Section II-C discusses the complexity of the motion planning problem for metamorphic robots.

A. Review of Lattice Kinematics

Consider $\mathbf{R^N}$ (N-dimensional Euclidean space) 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 $\{\vec{v}_i\}$ as follows:

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

A vast body of literature deals with the symmetry groups associated with lattices (which are simply discrete subgroups of $\mathbf{SE}(N)$ — the Special Euclidean Group[1]), and the decomposition of space into regular lattices, e.g., [15].

One way to view space is as a collection of connected close-packed polyhedra, the centers and/or vertices of which form a regular lattice. 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 collection of regular polyhedra which are either empty or filled. 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 which contain at least two lattice points (module centers), every point in the lattice is given a unique set of coordinates with the unit vectors $\{\vec{v}_i\}$ defining coordinate axes. While this coordinate system will generally be skewed, it will be a Cartesian system if the lattice has square or cubic spacing.

In order to define distance between configurations, we will first need a concept of distance between modules. The regular Euclidean metric is an acceptable choice but the one that more accurately reflects the least number of moves required by a module to move between two points is defined as follows. First construct a *lattice connectivity graph*, i.e., a graph with vertices at lattice points, and edges that are straight lines connecting all neighboring vertices. The distance measured along *shortest paths* connecting two lattice points in this graph is what we will refer to as the distance between two lattice points/modules. For example, if a metamorphic robot is composed of square or cubic modules, distance between modules would be given by the Manhattan/Taxicab metric in $\mathbf{R^N}$ (see [7] for explanation and other examples). We call this measure of distance a *lattice metric*, and denote it $\delta_L(a, b)$, where a and b are lattice points. By definition, the lattice metric yields the minimal distance between lattice points, while defining a path connecting all intermediate lattice points. This distance is a unique number, but the number of equidistant paths may be very large.

B. General Formulation of the Motion Planning/Reconfiguration Problem

In this section we formalize the motion planning problem for metamorphic robotic systems. Fig. 1 demonstrates the reconfiguration process with two planar hexagonal modules. 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 which are accessible and not already occupied.
- 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.
- At each timestep only one module may move, and this module may only move by one lattice space. It achieves this motion by deforming and mating faces to faces (or in the planar case edges to edges, as shown in Fig. 1).[2]

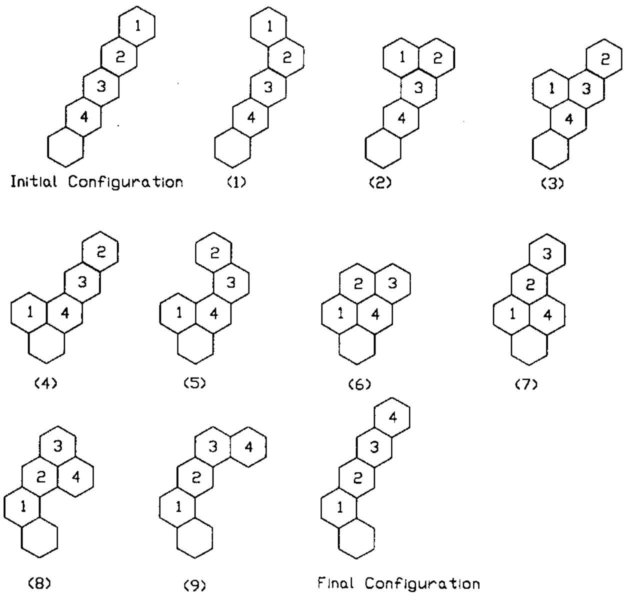


Fig. 3. A complete reconfiguration sequence involving two parallel configurations.

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.

Fig. 3 shows a complete reconfiguration sequence from one serial structure to another.

Observe that we consider only one module moving at any time. Two other motion strategies which can be employed are as follows. 1) Motion involving two or more modules moving together or separately at each time step without violating the connectivity constraints. This also includes *branches* of modules moving due to the motion of the module at the base of the branch. 2) *Fixed topology* motion in which the connection between the modules remains the same, and a change in configuration occurs by changing the joint angles of the modules. By focusing on single module motion, the restricted reconfiguration problem becomes tractable and helps to illustrate the metric concepts discussed in this paper (which are completely general since they do not assume any particular mode of reconfiguration). Furthermore, while single module motions are not always optimal, they are advantageous for obstacle avoidance and for motion in constrained spaces, and are therefore interesting in their own right.

[1] $\mathbf{SE}(N)$ is defined as the group of rigid motions, i.e., rotations and translations, in N-dimensional Euclidean space.

[2] This condition restricts the scope of the current work. Solution of this restricted problem is a starting point for treating the more general problem of multiple simultaneous module motions.

C. Complexity of the Motion Planning Problem

As described in Section II-B, the motion planning problem of metamorphic robots is defined as the reconfiguration of a collection of modules from an initial configuration to a final configuration based on certain constraints. This however leads to a computationally complex step of determining an optimal set of moves, i.e., the minimum number of moves required to completely reconfigure.

To the best of our knowledge, there is no simple method for solving the above problem. The reason is simply that for any number of modules n, the number of possible connected configurations is exponential in n. For a discussion of the complexity of this problem, see [8].

As a result, we have to look for heuristics which can give a near optimal solution. Any such heuristic requires a distance measure between configurations so that the shortest path between configurations is picked. The metrics discussed in the next section are some of the possible distance measures.

III. Defining Distance Between Configurations

In this section, we define measures of distance between *configurations* of any given metamorphic system as opposed to distance between *modules* as discussed in Section II-A. Each configuration of n modules is defined by the collection of n connected lattice spaces which it fills. That is, we do not distinguish between different modules and any permutation of labels has no effect on the configuration since all modules are identical. Therefore, two configurations with the same shape and relative position in space are said to be the same. Metrics that define distance between configurations in this way are denoted δ_C for "configuration metric."

Recall that a proper distance (or metric) function between points A and B in any given set (which for the problem at hand is the set of all connected configurations composed of n metamorphic robot modules) is defined by the following properties [23], [5], [10]:

$$d(A,B) \geq 0 \text{ and } d(A,B) = 0 \Longleftrightarrow A = B$$

$$d(A,B) = d(B,A) \tag{1}$$

$$d(A,B) + d(B,C) \geq d(A,C)$$

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*.

In the context of metamorphic robots A and B denote two configurations such that $a_i \in A$ represents a module in one configuration and $b_j \in B$ represents a module in another configuration for $i, j = 1, ..., n$.

A trivial example of a configuration metric is the *discrete metric*

$$\delta_C^{(0)}(A,B) = \begin{cases} 1 & A \neq B \\ 0 & A = B \end{cases} \tag{2}$$

which has a 0 value if the two configurations are identical and a value equal to 1 otherwise. Another illustration of a configuration metric is given in Fig. 3 which describes an optimal reconfiguration sequence. In this case the distance between any two configurations is the minimum number of moves required to reconfigure from one configuration to another. This metric is described in more detail in Section III-B.

We now discuss some of the configuration metrics which are used later for driving the reconfiguration process.

A. The Overlap Metric

One way of defining distance between two configurations is to consider the number of nonoverlapping modules in the two configurations. This represents the minimum number of modules which have to move for reconfiguration, but not the number of moves the modules make. This concept of distance is in fact a metric, denoted $\delta_C^{(1)}(A,B)$, which we call the Overlap Metric. This metric requires $O(n)$ computations, and is much more informative than $\delta_C^{(0)}$.

Theorem 3.1: The function

$$\delta_C^{(1)}(A,B) = n - |A \cap B| \tag{3}$$

which defines the number of nonoverlapping modules between any two configurations of modules is a metric.

Proof: In order to show that this is a metric we will use a few definitions from set theory: $|A| = |B| = n =$ number of elements in A and B; $A \cap B = \{x : x \in A \text{ and } x \in B\}$; $A \cup B = \{x : x \in A \text{ or } x \in B\}$; $A \backslash B = \{x : x \in A, x \notin B\}$. We now show that $\delta_C^{(1)}$ is a metric by showing that each of the properties in (1) hold.

1) Positive definiteness follows because $|A \cap B| \leq n$, and two configurations are said to be the same ($|A \cap B| = n$, or $\delta_C^{(1)}(A,B) = 0$) if and only if all modules overlap.
2) The symmetry condition in (1) follows from the fact that $|A \cap B| = |B \cap A|$.
3) Proof that the triangle inequality holds is as follows: Suppose we are given three configurations: A, B, and C (or equivalently the set of module locations that define the configuration). From set theory, we know that

$$\delta_C^{(1)}(A,B) = n - |A \cap B| = |A \backslash B| = |B \backslash A|$$

(even though $A \backslash B \neq B \backslash A$) because A and B have the same number of elements. We want to show that

$$\delta_C^{(1)}(A,C) \leq \delta_C^{(1)}(A,B) + \delta_C^{(1)}(B,C)$$

or equivalently,

$$|C \backslash A| \leq |B \backslash A| + |C \backslash B|.$$

We start by showing that

$$C \backslash A \subseteq (B \backslash A) \cup (C \backslash B). \tag{4}$$

Let x be an arbitrary element of $(C \backslash A)$. $x \in (C \backslash A)$ implies $x \in C$ and $x \notin A$. Furthermore, either $x \in B$ or it is not. These possibilities are written as

Case 1: $x \in B$ and $x \notin A$ means $x \in B \backslash A$;
Case 2: $x \notin B$ and $x \in C$ means $x \in C \backslash B$.

In either case $x \in (B\backslash A) \cup (C\backslash B)$ which proves (4). Therefore, $|C\backslash A| \leq |(B\backslash A) \cup (C\backslash B)| \leq |B/A| + |C/B|$ ■

While this is a valid metric on the set of all configurations of modules with the same number of modules, this metric fails to reflect the actual fewest number of moves needed to reconfigure from one configuration to another. In fact, no assumptions were made about the connectivity of the configurations or the type of overlap between configurations in the above proof.

B. The Minimal Number of Moves Metric

This subsection examines another metric on the set of all connected configurations of n modules.

Theorem 3.2: The function

$$\delta_C^{(2)}(A,B) = M_{\min}(A,B)$$

is a metric, where $M_{\min}$ is the fewest moves needed to reconfigure while observing locomotion constraints.

Proof:

1) *Positive Definiteness*: $\delta_C^{(2)}(A,B) \geq 0$, with equality only when there are zero moves required to reconfigure from one configuration to another, i.e., $A = B$.
2) *Symmetry*: $\delta_C^{(2)}(A,B) = \delta_C^{(2)}(B,A)$ because the minimal number of moves from one connected configuration to another can be performed in reverse order.
3) *Triangle Inequality*: Since $\delta(A,C)$ is defined to be the fewest moves required to reconfigure from A to C, any reconfiguration must require at least this many moves. Thus, a change to any intermediate configuration B and then from B to C must by definition observe $\delta_C^{(2)}(A,B) + \delta_C^{(2)}(B,C) \geq \delta_C^{(2)}(A,C)$. ■

Unfortunately, this metric has no representation other than explicitly solving a computationally explosive problem and recording the sum 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 remaining formulations of this section, and the optimal reconfiguration problem could simply be formulated as a shortest path problem on a graph where each edge is a move and each vertex is a configuration. But this is not possible due to the computational complexity of this approach.

C. The Optimal Assignment Metric $\delta_C^{(3)}$

In this subsection we define and illustrate one particular configuration metric called the *optimal assignment* metric, which is denoted as $\delta_C^{(3)}$. The distance $\delta_C^{(3)}(A,B)$ between two configurations A and B 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. Equivalently, this can be represented as a graph theory problem in which configurations A and B correspond to the two partite sets of a *bipartite graph*. The task then is to find a perfect matching in a weighted *bipartite graph* [11, 29] $G = (A,B)$, such that the sum of the weights of the matching is minimized. The weights correspond to the distance between two modules.

Section IV shows that the sum of distances between optimally assigned modules is in fact a metric on the set of all configurations of modules. Here we describe the optimal assignment problem and review an algorithm for solving it which is $O(n^3)$. This cost is still far less than the exponential order of computations required to compute $\delta_C^{(2)}$, but improves on $\delta_C^{(1)}$ by incorporating information about the distance between modules in the measure of distance between configurations.

1) Defining Optimal Assignment: Let m_{ij} be a variable which is 1 if module a_i in the present configuration maps to module b_j in the new configuration and 0 otherwise. $d_{ij} = \delta_L(a_i, b_j)$ is the lattice distance between module a_i and b_j. An arbitrary assignment will have an associated cost function

$$f(A,B) = \sum_{\substack{1\leq i\leq n \\ 1\leq j\leq n}} d_{ij} m_{ij} \tag{5}$$

with the constraints

$$\sum_{i=1}^{n} m_{ij} = 1 \qquad \text{for all} \;\; j = 1,..,n$$

and

$$\sum_{j=1}^{n} m_{ij} = 1 \qquad \text{for all} \;\; i = 1,..,n. \tag{6}$$

The constraints ensure that the mapping is a bijection. We define

$$\delta_C^{(3)}(A,B) = \min_{\Pi_n} f(A,B) \tag{7}$$

where Π_n is the set of all possible matchings. We will prove in Section IV that this definition satisfies the formal metric properties.

2) Evaluating Optimal Assignment: Several algorithms are available for solving this optimal assignment problem. The method described below is the Hungarian algorithm for optimal assignment [22].

Construct an $n \times n$ matrix $\mathbf{D}$, with elements $d_{ij} = \delta_L(a_i, b_j)$- the lattice distance between modules a_i and b_j. We wish to find an assignment m_{ij} in $\mathbf{D}$ which minimizes (5).

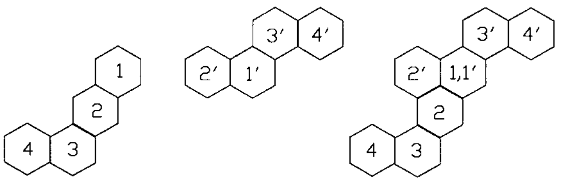
Observe that if we subtract a constant k_q from the q^{th} row of $\mathbf{D}$, giving rise to a new matrix $\mathbf{D'}$ with elements d'_{ij}, then

$$\sum_{\substack{1\leq i\leq n \\ 1\leq j\leq n}} d'_{ij} m_{ij} = \sum_{\substack{1\leq j\leq n \\ 1\leq j\leq n}} d_{ij} m_{ij} - k_q \sum_{1\leq j\leq n} m_{qj}$$

$$= \sum_{\substack{1\leq i\leq n \\ 1\leq j\leq n}} d_{ij} m_{ij} - k_q \tag{8}$$

using (6). Thus, an assignment m_{ij} that minimizes (5) also minimizes (8) and vice versa. The same result is obtained if a constant l_p is subtracted from the pth column. This gives us a method of finding the optimal assignment.

Let k_q be the minimum element in the qth row of $\mathbf{D}$ and l_p be the minimum element of the pth column. Subtract k_q



(a) (b) (c)

Fig. 4. (a) Present configuration. (b) New configuration. (c) Module labeling.

from each element of the qth row for all q. Subtract l_p from each element of the pth column for all p obtaining a matrix $\bar{\mathbf{D}}$. Let us call this the *reduced matrix*. This procedure produces at least one 0 in each row and column.

The problem can then be solved by finding an *independent set*[3] of n 0's in the reduced matrix. Note, that by finding an independent set of 0's we are essentially obtaining an assignment which has the minimal value or cost associated with it (i.e., 0 cost for the reduced matrix). Since the minimization problem remains the same, as shown in (8), the optimal assignment is simply given by taking m_{ij} equal to 1 for the i, j corresponding to the independent set of 0's. The maximum number of independent 0's can be found by using a corollary of the König–Egerváry theorem[4] [21].

Alternatively this is equivalent to finding an optimal matching in the *bipartite graph* $G = (A, B)$ where A and B represent the initial and final configurations and there's an edge between $a_i \in A$ and $b_j \in B$ iff $d'_{ij} \in \bar{\mathbf{D}} = 0$. If the number of independent 0's is equal to n then the solution is simply the assignment m_{ij} corresponding to the above 0's. Otherwise, we successively modify the reduced matrix to form a new *modified matrix* $\hat{\mathbf{D}}$ where there are n independent 0's.

One method to do this is to find out the minimum number of lines (one line refers to one complete row or column) which cover all the 0's in $\bar{\mathbf{D}}$. Let p be the smallest uncovered element. Modify the reduced matrix by subtracting p from all the uncovered elements and adding p to each twice covered element by the lines (i.e., each element which lies at the intersection of two lines). This is the modified matrix $\hat{\mathbf{D}}$. It is easy to show that the new modified matrix has been obtained from the preceding one by adding or subtracting a constant from different rows or columns. Thus, the minimization problem remains the same. The next step is to look for n independent 0's and if none are present, the process is repeated until an independent set is found.

The sum of d_{ij} corresponding to the matrix indexes i and j of n independent 0's constitutes the distance between two configurations using the optimal assignment metric $\delta_C^{(3)}$. The complexity of this optimal assignment algorithm is $O(n^3)$ where n is the number of modules. For a proof of this see [29]. Below we provide pseudocode which implements the above discussion.

Hungarian Algorithm for Computing the Optimal Assignment Metric

```
Assign $T = T_{initial}$
While {(Final Configuration not reached)
 and (moves made < moves allowed)}
 After every $k^{th}$ move, let $T = \beta \times T$
 Find the energy $E$ of the Current
 Configuration using the given cost
 function
 Find out all possible moves of all modules.
 For each possible move $i$
     Find the change in energy $\Delta E_i$, if that
     move is taken
 If there are moves for which $\Delta E$ is
 negative
     Pick any one of those moves
 Else If $\Delta E$ is positive for all moves
     Assign a probability
     $p_i = \frac{e^{-\Delta E_i/T}}{\sum_{i=1}^{k_m} e^{-\Delta E_i/T}}$
     to each move
     Pick a move based on the assigned
     probabilities.
```

In the following subsection, we illustrate this technique with an example.

D. An Example

For an illustration of the optimal assignment algorithm, consider the following example shown in Fig. 4.

The matrix $\mathbf{D}$ formed by the distances between various modules is shown in (9).

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1' & 2' & 3' & 4' \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 2 & 2 & 3 & 4 \\ 3 & 2 & 4 & 5 \end{pmatrix} \end{array}. \tag{9}$$

Performing column operations (subtracting l_p, the minimum element of each column, from each column, respectively), we get the matrix in (10). Similarly performing the row operations, we get the reduced matrix in (11).

$$\mathbf{D}' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 2 & 2 \\ 3 & 1 & 3 & 3 \end{pmatrix} \tag{10}$$

$$\bar{\mathbf{D}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 0 & 2 & 2. \end{pmatrix} \tag{11}$$

The next step is to modify the reduced matrix by subtracting the smallest element not covered by the lines (1 in the present case) from all the uncovered elements and by adding it to each twice covered element (i.e., lying at the intersection of two covering lines), Doing this we get the modified matrix $\hat{\mathbf{D}}$

[3] An independent set of 0's in a matrix is a set of 0's, no two of which are in the same row or same column.

[4] König–Egerváry theorem: If $\mathbf{D}$ is a matrix of 0's and 1's, a maximum independent set of 0's has the same number of elements as a minimum set of lines covering all the 0's of $\mathbf{D}$. Corollary: The number of independent 0's in the *reduced or modified matrix* is equal to the minimum number of lines which cover all 0's.

as shown in (12). This matrix contains several combinations of four independent 0's any of which solves the problem and gives the value of $\delta_C^{(3)}(A,B)$.

$$\hat{\mathbf{D}} = \begin{pmatrix} \boxed{0} & 1 & 0 & 0 \\ 0 & 0 & 0 & \boxed{0} \\ 0 & 0 & \boxed{0} & 0 \\ 1 & \boxed{0} & 1 & 1. \end{pmatrix} \tag{12}$$

Choosing the boxed solution above, the value of $\delta_C^{(3)}(A,B)$ is given as

$$\delta_C^{(3)}(A,B) = d_{11'} + d_{24'} + d_{33'} + d_{42'} = 0+3+3+2 = 8.$$

The minimal value is achieved by matching modules with the subscripts in the above expression. The reader is encouraged to verify this by trying to assign modules from each configuration in Fig. 3, and summing the lattice distances between all of them.

IV. Metric Properties of Optimal Assignment

In Section IV-A, we prove that $\delta_C^{(3)}(.,.)$ is a metric. In Section IV-B, we use this fact to show that the optimal matching approach only needs to consider pairings of *nonoverlapping* modules, thus reducing the size of the matrices generated using the techniques of Section III.

A. The Optimal Assignment Metric

In this subsection we show that $\delta_C^{(3)}$ is a metric. We begin with some background material. Let $\{a_1, a_2, \cdots, a_n\}$ denote the modules of A and $\{b_1, b_2, \cdots, b_n\}$ denote the modules of B, where A and B represent two configurations. An assignment from A to B is a *bijective* function from the members of A to the members of B. It can be represented by a function on the indexes $\{1, 2, \cdots, n\}$ of A to the indexes $\{1, 2, \cdots, n\}$ of B. Functions of this kind are permutations:

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix}.$$

For n modules there exist exactly $n!$ different permutations, i.e., $n!$ different ways to rearrange the numbers in the set $\{1, 2, \ldots, n\}$. We list here two properties of permutations which are needed later on. For more details refer to [30].

- Π_n is a group with operation $\circ$, which is the composition of two permutations:

$$(\pi_1 \circ \pi_2)(i) = \pi_1(\pi_2(i)) \in \Pi_n \qquad \forall \pi_1, \pi_2 \in \Pi_n.$$

- Because Π_n is a group, each permutation $\pi \in \Pi_n$ has a unique inverse element $\pi^{-1} \in \Pi_n$, such that

$$\pi \circ \pi^{-1} = I = \pi^{-1} \circ \pi$$

where I is the unique identity element.

A permutation π applied to two configurations A, B yields the assignment $\hat{\pi}_{AB}$:

$$\hat{\pi}_{AB} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_{\pi(1)} & b_{\pi(2)} & \cdots & b_{\pi(n)} \end{pmatrix}.$$

We are interested in the sum of the distances of the matched module pairs, which are given by the columns of $\hat{\pi}_{AB}$:

$$f_{\hat{\pi}}(A,B) = \sum_{i=1}^{n} d(a_i, b_{\pi(i)})$$

where d is any metric between two modules, but in particular the lattice metric, i.e., unless otherwise specified $d(a,b) = \delta_L(a,b)$.

For later consideration it is important to note that this sum remains the same if we rearrange the module pairs in the following way:

$$\sum_{i=1}^{n} d(a_i, b_{\pi(i)}) = \sum_{i=1}^{n} d(a_{\varphi(i)}, b_{\pi(\varphi(i))}) \tag{13}$$

where $\varphi \in \Pi_n$ is an arbitrary permutation. This is true, since φ is applied directly to the index i, which means that it only changes the order of appearance of the terms in the sum.

From all possible permutations we take the one that gives us the minimal value (optimal assignment), i.e., we use the definition for $\delta_C^{(3)}$ as given in Section III-A.

Theorem 4.1: $\delta_C^{(3)}$ is a metric on the set of all possible configurations, i.e., it is 1) positive definite, 2) symmetric, and 3) the triangle inequality holds.

Proof: For explanations of the steps in 2) and 3) please refer to the numbered remarks after the transformations.

1) *Positive definiteness*:

$$\delta_C^{(3)}(A,B) = \min_{\pi \in \Pi_n} \left(\sum_{i=1}^{n} \underbrace{d(a_i, b_{\pi(i)})}_{\geq 0} \right) \geq 0$$

$$\delta_C^{(3)}(A,B) = 0 \quad \Rightarrow \quad d(a_i, b_{\pi(i)}) = 0 \forall i \Rightarrow A = B.$$

2) *Symmetry:*

$$\begin{aligned} \delta_C^{(3)}(A,B) &= \min_{\pi \in \Pi_n} \sum_{i=1}^{n} d(a_i, b_{\pi(i)}) \\ &\overset{(1)}{=} \min_{\pi \in \Pi_n} \sum_{i=1}^{n} d(a_{\pi^{-1}(i)}, b_i) \\ &\overset{(2)}{=} \min_{\pi \in \Pi_n} \sum_{i=1}^{n} d(b_i, a_{\pi^{-1}(i)}) \\ &\overset{(3)}{=} \min_{\varphi \in \Pi_n} \sum_{i=1}^{n} d(b_i, a_{\varphi(i)}) \\ &= \delta_C^{(3)}(B,A). \end{aligned}$$

i) We use (13) with $\varphi = \pi^{-1}$.

ii) The distance function $d(\cdot,\cdot)$ is a metric and therefore symmetric.

iii) Minimizing over all $\pi \in \Pi$ is the same as minimizing over all $\pi^{-1} \in \Pi$.

3) Considering the terms on the right-hand side of the inequality:

$$
\begin{aligned}
&\delta_C^{(3)}(A,B)+\delta_C^{(3)}(B,C)\\
&= \min_{\varphi\in\Pi_n}\sum_{i=1}^{n} d(a_i,b_{\varphi(i)}) \quad + \quad \min_{\psi\in\Pi_n}\sum_{i=1}^{n} d(b_i,c_{\psi(i)})\\
&\overset{(1)}{=} \min_{\varphi\in\Pi_n}\min_{\psi\in\Pi_n}\left(\sum_{i=1}^{n} d(a_i,b_{\varphi(i)})+\sum_{i=1}^{n} d(b_i,c_{\psi(i)})\right)\\
&\overset{(2)}{=} \min_{\varphi\in\Pi_n}\min_{\psi\in\Pi_n}\left(\sum_{i=1}^{n} d(a_i,b_{\varphi(i)})+\sum_{i=1}^{n} d(b_{\varphi(i)},c_{\psi(\varphi(i))})\right)\\
&= \min_{\varphi\in\Pi_n}\min_{\psi\in\Pi_n}\sum_{i=1}^{n}\underbrace{\left(d(a_i,b_{\varphi(i)})+d(b_{\varphi(i)},c_{\psi(\varphi(i))})\right)}_{\geq d(a_i,c_{\psi(\varphi(i))})}\\
&\geq \min_{\varphi,\psi\in\Pi_n}\sum_{i=1}^{n} d(a_i,c_{\psi\circ\varphi(i)})\\
&= \min_{\pi\in\Pi_n}\sum_{i=1}^{n} d(a_i,c_{\pi(i)})\\
&= \delta^{(3)}(A,C).
\end{aligned}
$$

1) We use the following equality:

$$\min_{\varphi\in\Pi_n} f(\varphi)+\min_{\psi\in\Pi_n} g(\psi)=\min_{\varphi\in\Pi_n}\min_{\psi\in\Pi_n}\,(f(\varphi)+g(\psi)).$$

2) We use (13) for the sum at the right. ■

B. Reduction of the Computational Cost

In the previous subsection we saw that $\delta_C^{(3)}(A,B) = \min_{\pi\in\Pi_n}\sum_{i=1}^{n} d(a_i,b_{\pi(i)})$ is a metric on the set of all configurations, defining the distance between any two configurations A and B. In this subsection we prove that we can restrict our search to permutations which match only the nonoverlapping modules of two configurations to each other thereby reducing the size of the matrices considered in the Hungarian algorithm in the previous section. This reduces the computational effort considerably in most cases. Note, however, that optimal assignments exist which do not assign overlapping modules to each other. Fig. 5 shows the configurations corresponding to the example of Section III-D. Fig. 5(a)–(c) are three possible optimal assignments. While (a) assigns the overlapping modules to each other, (b) and (c) do not but are still valid optimal assignments.

In the proof which follows, the notation $a \sim b$ is used to indicate that module $a \in A$ is matched to module $b \in B$ by assignment π^*, whereas $a \not\sim b$ means that they are not.

Theorem 4.2: An optimal assignment between two configurations A and B can always be obtained by considering only the nonoverlapping modules and assigning the overlapping modules to each other.

Proof: Consider an optimal assignment π^*, in which not all overlapping modules are matched to their counterparts. Without loss of generality, let $a_1 \in A$ be a module in the overlap of A and B, which is not matched to its counterpart $b_1 \in B$, i.e.,

$$a_1 = b_1, \quad \text{but} \quad a_1 \not\sim b_1.$$

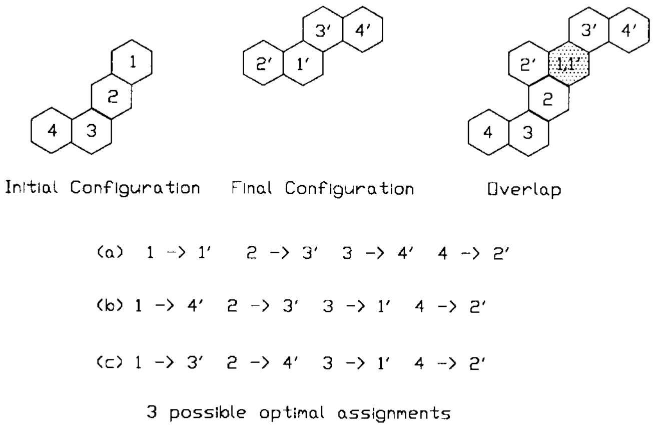

1
2
4
3
3'
4'
2'
1'
3'
4'
2'
1,1'
2
4
3
Initial Configuration
Final Configuration
Overlap
(a) 1 -> 1' 2 -> 3' 3 -> 4' 4 -> 2'
(b) 1 -> 4' 2 -> 3' 3 -> 1' 4 -> 2'
(c) 1 -> 3' 2 -> 4' 3 -> 1' 4 -> 2'
3 possible optimal assignments

Fig. 5. Three possible optimal assignments for the given initial and final configurations.

Let $b_2 \in B$ be the module to which a_1 is matched, and $a_2 \in A$ be the module to which b_1 is matched,

$$a_1 \sim b_2, \quad \text{and} \quad a_2 \sim b_1.$$

Consider a modified assignment π', where a_1 is assigned to b_1, $a_1 \sim b_1$, and a_2 is assigned to b_2, $a_2 \sim b_2$. Comparing the expression $f_{\pi'}(A,B)=\sum_{i=1}^{n} d(a_i,b_{\pi'(i)})$ with the minimal value $f_{\pi^*}(A,B)=\sum_{i=1}^{n} d(a_i,b_{\pi^*(i)})$ we note that

$$f_{\pi'}(A,B)=f_{\pi^*}(A,B)+T_2-T_1$$

where

$$T_1 = d(a_1,b_2)+d(a_2,b_1) \qquad T_2 = d(a_1,b_1)+d(a_2,b_2).$$

T_2 is smaller or equal to T_1, since

$$
\begin{aligned}
T_2 &= \underbrace{d(a_1,b_1)}_{0}+d(a_2,b_2)=d(a_2,b_2)\\
&\leq d(a_2,b_1)+d(b_1,b_2)\\
&\leq d(a_2,b_1)+\underbrace{d(b_1,a_1)}_{0}+d(a_1,b_2)\\
&= d(a_2,b_1)+d(a_1,b_2)\\
&= T_1.
\end{aligned}
$$

Hence we have

$$f_{\pi'}(A,B)\leq f_{\pi^*}(A,B).$$

But since π^* is already an optimal assignment, $f_{\pi'}(A,B)$ can not be smaller than $f_{\pi^*}(A,B)$ and therefore both terms must be equal,

$$f_{\pi'}(A,B)=f_{\pi^*}(A,B)$$

and π' is also an optimal assignment.

We can apply this procedure to the resulting modified assignment(s), until no such a_1, b_1 can be found. The result will be a minimal permutation where all modules in the overlap are matched to their counterparts. ■

Hence it is possible to restrict our search for a minimal assignment to permutations of nonoverlapping modules right from the beginning. In cases where there is substantial overlap, this can save a lot of of computational effort.

V. Modified Metric Functions

In the previous sections we defined some basic metric functions which are useful for modular robot motion planning. In this section we consider how existing metrics can be combined to form new metrics. This is motivated by the fact that a function which combines the properties of the metrics discussed earlier when used with a proper weighting can yield better results than the original metrics.

We first prove some basic theorems.

Theorem 5.1: If δ_1 and δ_2 be two metrics, then $\delta_3 = \alpha\delta_1 + \beta\delta_2$ is also a metric, where α and β are fixed positive real numbers

Proof: Since δ_1 and δ_2 are metrics, we have:

1) *Positive Definiteness*: $\delta_3(A,B) = \alpha\delta_1(A,B) + \beta\delta_2(A,B) > 0, A \neq B$, $\delta_3(A,A) = \alpha\delta_1(A,A) + \beta\delta_2(A,A) = 0$;
2) *Symmetry*: $\delta_3(A,B) = \alpha\delta_1(A,B) + \beta\delta_2(A,B) = \alpha\delta_1(B,A) + \beta\delta_2(B,A) = \delta_3(B,A)$;
3) *Triangle Inequality*: $\delta_3(A,B) + \delta_3(B,C) = \alpha\delta_1(A,B) + \alpha\delta_1(B,C) + \beta\delta_2(A,B) + \beta\delta_2(B,C) \geq \alpha\delta_1(A,C) + \beta\delta_2(A,C) \geq \delta_3(A,C)$. ■

Theorem 5.2: If $f(\cdot,\cdot)$ satisfies all the properties of a metric except $f(A,A) \neq 0$, then the function $g(A,B) = f(A,B)\delta_C^{(0)}(A,B)$ is a metric where $\delta_C^{(0)}(\cdot,\cdot)$ is the discrete metric defined in (2).

Proof: Symmetry holds because both $f(\cdot,\cdot)$ and $d_0(\cdot,\cdot)$ are symmetric. The triangle inequality is unchanged for $A \neq B \neq C$. Likewise, $g(A,B) > 0$ if $A \neq B$. In fact, the only thing that is changed is that now $g(A,A) = 0$. ■

We are now ready to define two new metrics using the properties discussed above. These new metrics are useful because in some scenarios they improve performance in the motion planning problem.

A. Modified Overlap and Optimal Assignment Metric $\delta_C^{(4)}$

Using the overlap and the optimal assignment metrics discussed in the previous sections and using the theorems at the beginning of this section, we define a new metric $\delta_C^{(4)}$ as

$$\delta_C^{(4)}(A,B) = \alpha\delta_C^{(1)}(A,B) + \beta\delta_C^{(3)}(A,B). \tag{14}$$

The above metric is of interest since for some reconfigurations it is desirable to keep the overlapping modules in the two configurations in place, while for other cases it is desirable to move the overlapping modules. These preferences can be achieved by changing the values of α and β.

B. Configuration Metric $\delta_C^{(5)}$

Using Theorems 5.1 and 5.2, a new lattice metric $\delta_K(a,b)$ is defined from an old one as

$$\delta_K(a,b) = \alpha\delta_L(a,b) + \beta d_0(a,b)[\delta_L(a,0) + \delta_L(b,0)]. \tag{15}$$

A new configuration metric $\delta_C^{(5)}$ is then defined as

$$\delta_C^{(5)}(A,B) = \min_{\pi\in\Pi_n} \sum_{i=1}^{n} \delta_K(a_i, b_{\pi(i)}). \tag{16}$$

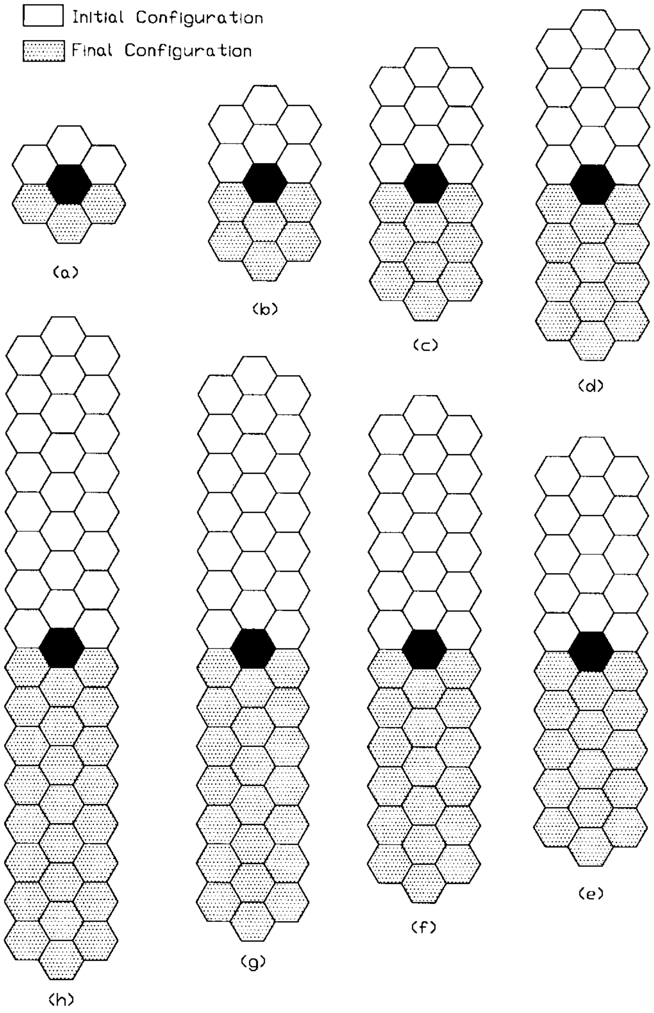


Fig. 6. Eight configurations used for evaluating real time taken.

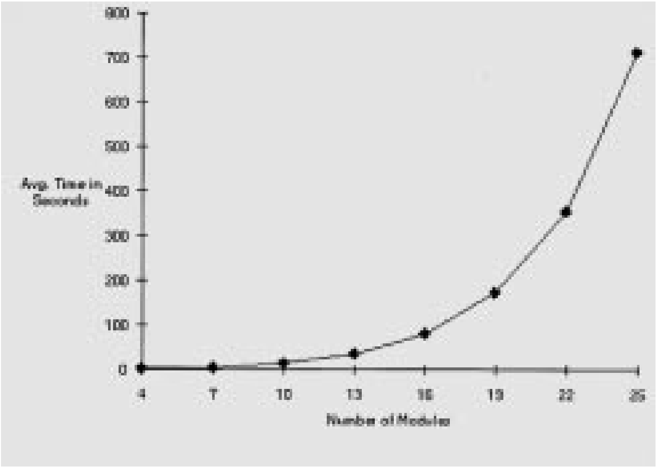


Fig. 7. Plot showing the average time taken to reconfigure for different number of modules.

The above definition of $\delta_C^{(5)}(A,B)$ is motivated by the fact that quite often we have two configurations next to each other or parallel to each other (as shown in Fig. 8). A pure optimal assignment based on the lattice metric δ_L simply assigns the modules next to each other and hence is not reflective of

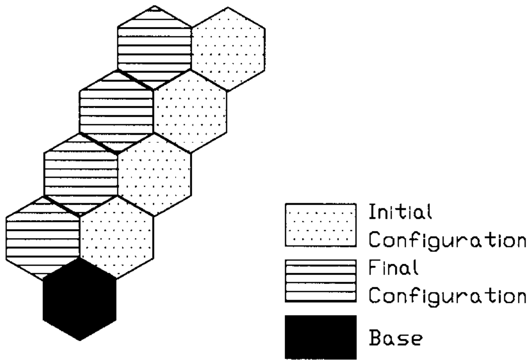


Fig. 8. Reconfiguration involving two serial structures parallel to each other.

the actual moves made by the modules. Whereas, the above approach tries to assign modules farthest from the base (0 position) in one configuration to the modules closer to the base in the other configuration.

In order to demonstrate the usefulness of these metrics, they will be used together with the method of Simulated Annealing, which drives the distance between two arbitrary configurations to zero by minimizing the "distance" between them. In fact, any method of discrete optimization could be used in place of Simulated Annealing. We use SA here because of its generality and ubiquity in the literature.

VI. The Method of Simulated Annealing

In this section we use the metrics discussed in the previous sections in a reconfiguration algorithm based on simulated annealing.

A. Simulated Annealing

Simulated Annealing is an algorithmic approach to solving optimization problems especially in cases where the global extremum is hidden among several local extrema [20]. The basic idea behind this algorithm comes from an analogy with simulating the annealing of solids [24] and slow cooling of liquids, i.e., the way metals or crystals cool and anneal to achieve the minimum energy state.

The basic simulated annealing algorithm considers the objective function to be minimized as the energy of the system. Starting from an initial state with energy E the system is perturbed to a neighboring state and the change in energy ΔE computed. If ΔE is negative, i.e., the energy is less in the new state, then the new state is accepted. If ΔE is positive, then the new state is accepted with a probability usually taken as $e^{-\Delta E/T}$, where T is a control parameter corresponding to temperature in the analogous case of thermodynamic cooling. In addition to the *energy function* and the *control parameter* T, a *cooling schedule* is required, i.e., a scheme for changing T as the algorithm proceeds, usually taken as $T_{i+1} = \beta \times T_i$ where β is a constant. Initially T is set to a high value and after a certain number of steps (k_c) at each value of T, its value is decreased by the factor β. Finally a *stopping criterion* is required to end the algorithm.

B. Energy Functions for Simulated Annealing

In the application of simulated annealing to metamorphic robot reconfiguration, energy functions which reflect the differences between configurations are important. Using measures of distance that formally satisfy the definition of a metric guarantee that we have a well defined stopping criterion, i.e., $\delta_C(A_g, A_k) = 0$ or $k = M_{max}$ (which ever comes first), where A_k is the k^{th} configuration in a sequence of configurations, A_g is the goal configuration, and M_{max} is the maximum allowable moves. Furthermore, the triangle inequality is important because if we can find a configuration A_i such that $\delta_C(A_o, A_g) = \delta_c(A_o, A_i) + \delta_c(A_i, A_g)$ and $\delta_C(A_o, A_i) \approx \delta_C(A_i, A_g)$ then the reconfiguration problem can be divided into two problems, which are likely to converge much faster.

Before discussing the energy functions we used in trial runs, consider a few naive choices of energy which were *not* used. Let C be the current configuration and F be the final configuration, where $C = \{c_i\}$ and $F = \{f_i\}$ are collections of lattice points (modules) representing the two configurations. *Any* energy function will be a function of C and F. As a first naive choice, consider:

$$E'(C, F) = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_L(c_i, f_j).$$

This is simply the sum of lattice distances of every module in one configuration to every module in the other. This function observes the triangle inequality when evaluated with arbitrary configurations, i.e., $E'(A, C) \leq E'(A, B) + E'(B, C)$. This follows from the fact that $\delta_L(a_i, b_j) + \delta_L(b_j, c_k) \geq \delta_L(a_i, c_k)$; summing over i, j, k and dividing by n, we get the triangle inequality for $E^{'}$. However, this function does not satisfy positive definiteness: $E'(A, A) \neq 0$. Furthermore, it is possible for $E'(C, F) = E'(F, F)$ without $C = F$. In other words, it is possible for minimization of this cost to not lead to the goal, and not even know when it has reached the goal.

This flaw in $E'(\cdot, \cdot)$ can be repaired by defining the following:

$$E''(C, F) = \delta_C^0(C, F) E'(C, F)$$

where δ_C^0 is the discrete configuration metric defined earlier. But in doing so, the configurations can still be close to each other (but not equal) without this being reflected in the distance function $E''(\cdot, \cdot)$. That is, there is not a gradual descent of $E^{''}$ to zero as $C \rightarrow F$, but $E^{''}$ is relatively "flat" until $C = F$.

In this paper we use four types of energy functions based on metrics instead of intuition, i.e., all are of the form:

$$E(i) = \delta_C^{(i)}(C, F)$$

for $i = 1, 3, 4, 5$, where $\delta_C^{(i)}(\cdot, \cdot)$ are the configuration metrics discussed in the earlier sections of this paper.

C. Reconfiguration Algorithm based on Simulated Annealing

In our implementation of simulated annealing, the change in energy associated with changing from any current configuration to all possible neighboring configurations is computed at each step in the algorithm. A neighboring configuration of C (denoted C_N) is a connected configuration which can be obtained by one move of any of the modules. If a move, or moves results in a neighboring configuration with reduced energy, then one of these moves is selected randomly. If none of the moves result in a decrease in energy, then a normalized probability is obtained for each move based on the probability function

$$p_i = \frac{e^{-\Delta E_i/T}}{\sum_{i=1}^{k_m} e^{-\Delta E_i/T}}$$

where k_m is the number of all possible moves.

A move is then picked based on its probability. The algorithm can be described as shown below.

Reconfiguration Using Simulated Annealing

In principle, this algorithm is easily changed to address the issue of multiple module motions, whether these are branch motions or multiple simultaneous single module motions. However, in practice to solve these problems in an optimal or near-optimal way, the set of all possible moves would have to be generated by partitioning the configuration into all possible collections of modules and each group of modules moved according to the constraints to check if the energy is reduced. The problem with this approach is that there are an exponential number of partitions of a given configuration (2^n to be exact [30]), and the implementation would therefore be problematical. Instead of addressing this issue further, and digressing from the main subject of this paper (which is the usefulness of the metrics proposed earlier) we leave the development of heuristics for the multiple module motion case for future work.

D. Time Complexity

As discussed in Section II-C, the problem of determining the minimal number of moves for metamorphic robots to completely reconfigure is computationally complex. A brute force method for finding an optimal solution is extremely complicated and time consuming because: 1) The number of possible configurations for any given number of modules n is exponential[5]; 2) There is not even a well-defined method for enumerating these configurations, let alone searching a graph with such configurations as vertices. Simulated Annealing offers a fast way of computing a near optimal solution by performing a number of offline trials and picking the best one out of them for actual reconfiguration. As a result the solution can be obtained very quickly. A plot of the average time taken on a Pentium-60 computer over a set of 20 trials for eight similar configurations (shown in Fig. 6) using optimal assignment metric $\delta_C^{(3)}(C, F)$ as the energy function is shown in Fig. 7.

As can be seen, even with as many as 25 modules the method works on the order of 10 min for each trial on a PC. Using direct evaluation of optimal assignment clearly has limitations since it requires $O(n^3)$ computations in the worst case. However, this is dramatically better than exhaustive graph search.

For the case when hundreds or thousands of modules are involved, further efficiencies can be gained using optimal assignment because on average an increase in the number of modules means the number of single module motions in a given configuration that can reduce energy increase. This means that optimal assignment need not be evaluated after each move, but rather after all individual energy decreasing moves of a given configuration are made. Furthermore, in the case when n is large, the optimal assignment need not involve all modules. That is, near optimal move sequences can be obtained by optimally assigning subsets of modules in two configurations while considering the other parts of the configurations as fixed. If the subsets of modules in two configurations that are assigned consist of only a constant (but large) number of modules, then a constant order of computation is required to evaluate optimal assignment, and the time plot in Fig. 8 would grow much more slowly for very large n.

E. Improving Simulated Annealing

A basic simulated annealing algorithm is prone to oscillations which slow down the convergence of the algorithm. To prevent such oscillations and to increase the efficiency, moves leading back to the previous configuration in a sequence of configurations can be disallowed. Of course, this restriction could be extended such that no configuration is allowed twice in a given sequence of moves, but this does not appear to be as common of an occurrence. If required, this can be achieved by simply removing all configurations between any two occurrences of the same configuration. These are loops in the configuration space of the metamorphic system that do not contribute toward attaining the goal. Other methods for improving the performance of a given heuristic can be found in [8].

VII. Results

We ran twenty trials of the simulated annealing algorithm for three sets of initial and final configurations (Figs. 8, 10, and 12) for each of eight different initial values of T, namely $T = 5$, 10, 20, 50, 100, 500, 1000. Four different energy functions discussed in Section VI were used for each of the configurations. Different values of α and β were tried for the energy functions $E(4)$ and $E(5)$ as shown in the the latter part of this section and the best results used in Figs. 9, 11, and 13. The *annealing schedule* consisted of 10 moves at each value of T followed by a decrease in the value of T by a factor of 0.8, i.e., $T_{i+1} = 0.8 * T_i$. The algorithm stopped if the final configuration was reached or if 300 moves had taken place. The results for the three cases are shown in Fig. 9.

Three typical sets of configurations were chosen in order to ascertain the behavior of the simulated annealing algorithm with four different energy functions. Fig. 9 shows the result for configurations corresponding to Fig. 8. The initial and final configurations in this case are two serial structures parallel to each other. In addition to the information in the graphs, it

[5] Only asymptotic results for the number of possible configurations are available in the literature.

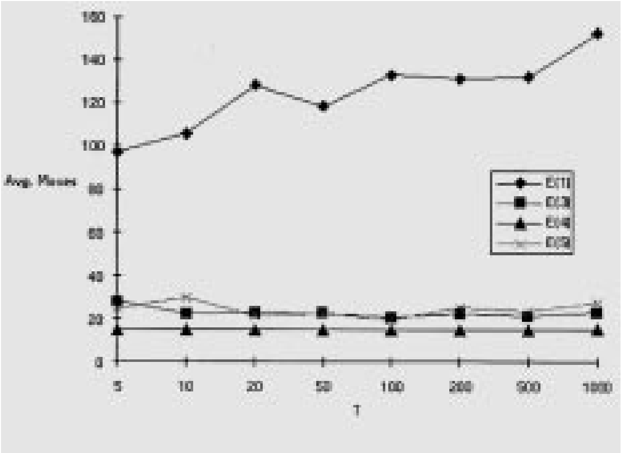


Fig. 9. Results for the serial configuration.

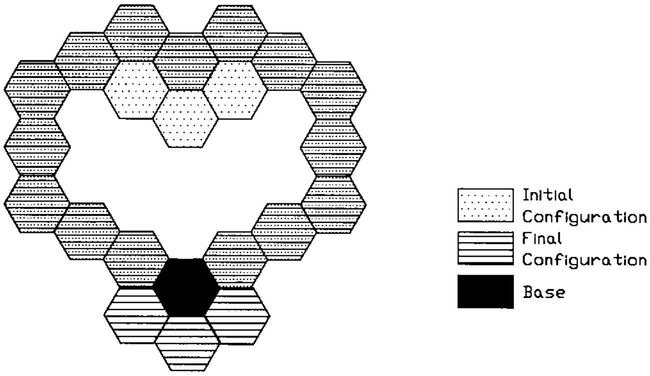


Fig. 10. Reconfiguration involving breaking a loop structure.

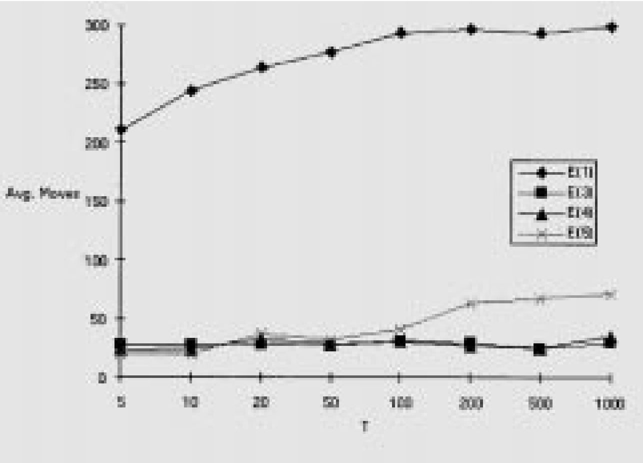


Fig. 11. Results for the loop configuration.

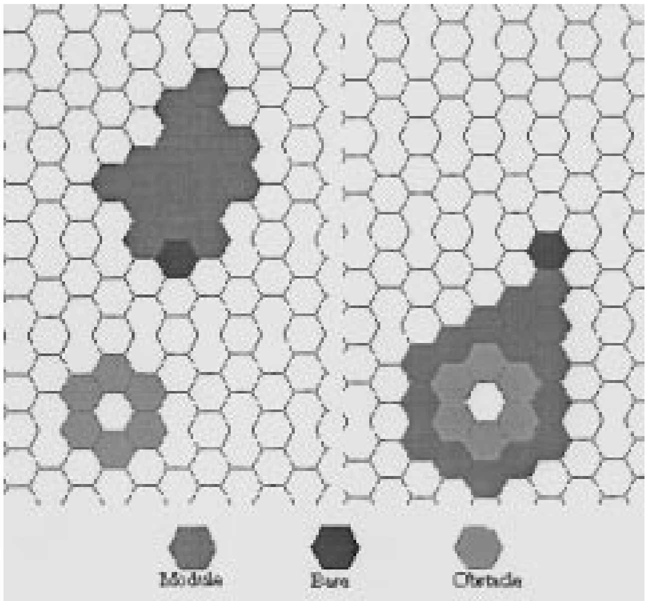


Fig. 12. Initial and final configurations for reconfiguration involving obstacle envelopment.

is interesting to note that actual fewest moves in this case is 10, and the best result generated by simulated annealing is 15. In fact, the serial case is the worst case for simulated annealing. Because of the motion constraints, the modules cannot simply move into the lattice spaces corresponding to the final configuration, but instead have to move over each other to attain the final configuration. Hill climbing is involved in this case and the three energy functions ($E(3) = \delta_C^{(3)}; E(4) = \delta_C^{(4)}, \alpha = 50, \beta = 1; E(5) = \delta_C^{(5)}, \alpha = 50, \beta = 1$) involving optimal assignment in some form yield much better results than the simple overlap function ($E(1) = \delta_C^{(1)}$). Observe that the value of α used for $E(4)$ and $E(5)$ corresponds to the best results obtained in Fig. 14 for different α values.

The results in Fig. 11 correspond to the configurations in Fig. 10. In this case, the best results obtained by simulated annealing is 14 moves, which is in fact the minimal number. In this case the overlapping modules form a loop. The nonoverlapping modules corresponding to the initial configuration lie inside the loop while those corresponding to the final configuration lie outside the loop. Since both energy functions can be locally minimized by preserving the overlap, a definite hill climbing is involved. Again, the three energy functions ($E(3) = \delta_C^{(3)}; E(4) = \delta_C^{(4)}, \alpha = 50, \beta = 1; E(5) = \delta_C^{(5)}, \alpha = 50, \beta = 1$) involving optimal assignment in some form yield much better results than the simple overlap function ($E(1) = \delta_C^{(1)}$). Since the existing overlap needs to be broken in order to reconfigure, the overlap metric performs poorly.

The third case involves reconfiguration in the presence of obstacles as shown in Fig. 12. The modules cannot move into the spaces occupied by the obstacles in the lattice space but have to move around the obstacles to envelop them. Fig. 13 shows the result for obstacle envelopment. The best sequence of moves that we were able to construct by hand required 110 moves, and the best simulated annealing trial achieved reconfiguration in 121 moves.

In this case, the metamorphic system is not able to reconfigure at all using the overlap metric $\delta_C^{(1)}$ as energy function and the simulation is terminated when 300 moves are over. The average number of moves made for the energy functions involving optimal assignment is much larger in this case because of branching. Since the final configuration is distributed around the obstacle, different branches of modules

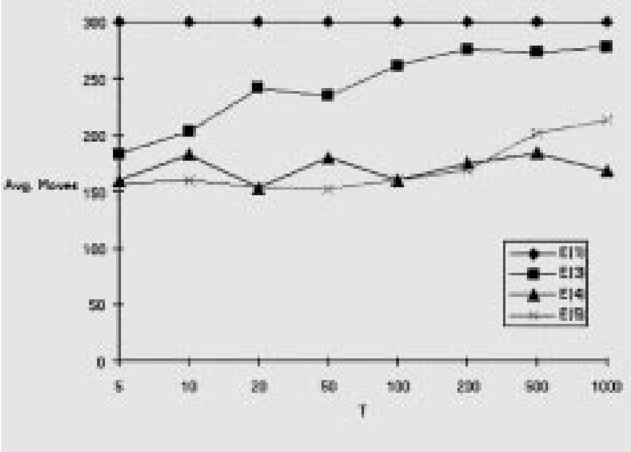
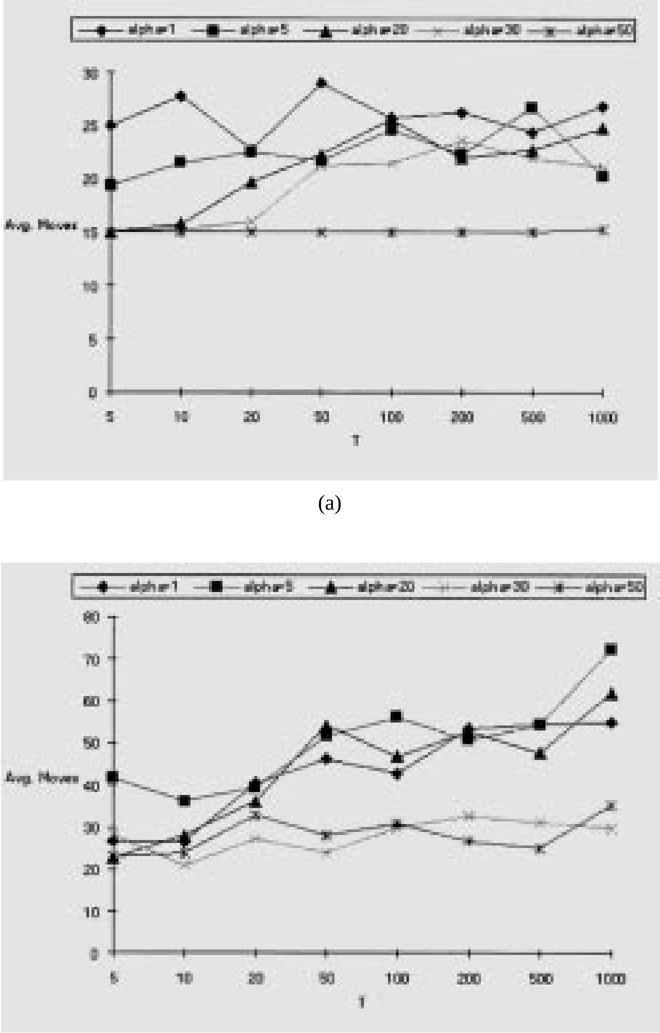


Fig. 13. Results for reconfiguration involving obstacle envelopment.

sprout from the initial configuration in order to minimize the energy. However, once a local minimum is reached, the modules in the branches have to climb steep hills because of connectivity constraints. As a result, the number of moves needed to reconfigure becomes large. The results for energy functions $E(4)$ and $E(5)$ are relatively better than pure optimal assignment since these functions try to preserve the overlap to an extent and hence avoid unnecessary oscillations when local minima occur.

As can be seen, the energy function corresponding to the optimal assignment metric $\delta_C^{(3)}$ and the functions which incorporate optimal assignment yield better result than the overlap metric in all cases. When the former is used, the moves made are usually those which reduce the distance between an empty lattice point in the final configuration and a module in the present configuration. In the case of the latter unless there is a move which increases the overlap, both good and bad moves are equally likely. Also, by incorporating the features of both the overlap metric and the optimal assignment metric, as is the case for energy functions E(4) and E(5), better performance can be obtained for most reconfiguration processes.

Another observation was that the initial temperature had no noticeable effect when the optimal assignment metric is used as the energy function except for the reconfiguration involving obstacle envelopment in Fig. 12. This is because if there is a move possible which reduces energy, simulated annealing will always choose that and in that case the value of the ratio $\Delta E/T$ does not influence the result. For example in this case ΔE_i is always negative for some move i, until a local minima is reached and such minima are few in the complete reconfiguration of the robot from the initial to the final configuration. Hence the above behavior. In the case when the overlap metric is used as an energy function there are a large number of local minima and plateaus, i.e., there's no move which decreases energy. As a result when T is large the value of the ratio $\Delta E/T$ is approximately the same for all moves. This results in an approximately equal probability for all moves. And so a bad move is as likely as a good move. This affects the average number of moves required to reconfigure. For the case involving obstacle envelopment, a high initial value of T leads to an increased branching effect and thus increases the average number of moves needed to reconfigure.



(a)

(b)

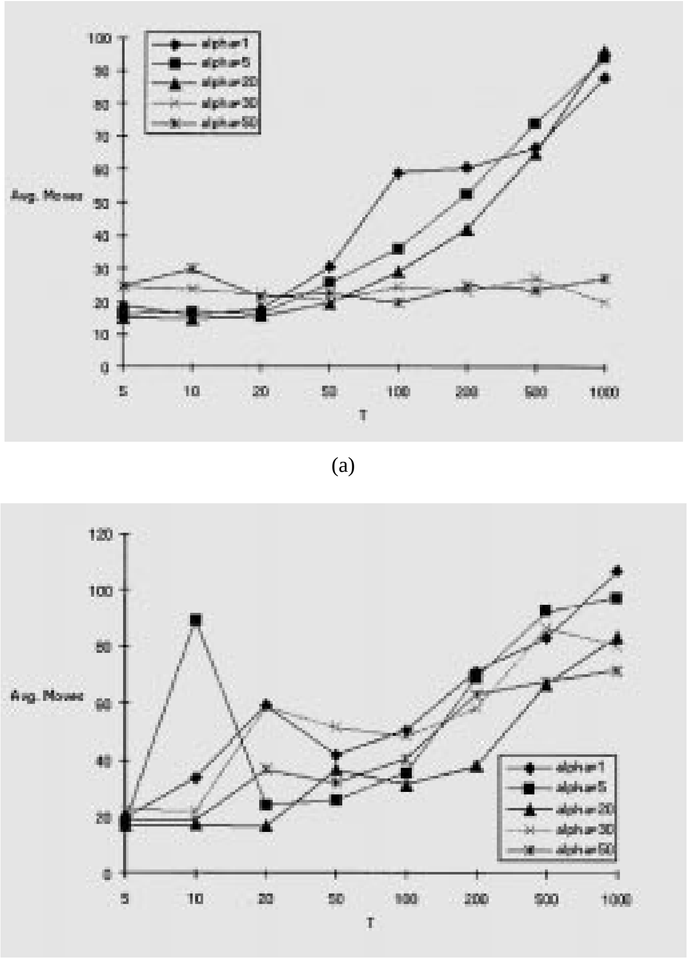
Fig. 14. Results for reconfiguration involving (a) serial configuration and (b) loop configuration using $E(4)$ for different α values.

We also ran simulations for different values of α for the energy functions $E(4) = \alpha\delta_C^{(1)}(C,F) + \beta\delta_C^{(3)}(C,F)$ and $E(5) = \delta_C^{(5)}(C,F)$ with $\beta = 1$ for the configurations shown in Figs. 8 and 10.

For the case of $E(4)$, the results are shown in Fig. 14(a) and (b) for five different α values, α = 1, 5, 20, 30, 50. The best results were obtained with a high value of α. A large weight on the overlap metric drove the reconfiguration process toward maintaining the overlap and thus avoiding oscillations. The results stabilized for $\alpha = 50$ and higher values for the case of Figs. 8 and 10.

The results for $E(5)$ are shown in Fig. 15(a) and (b) for five different α values, $\alpha =$ 1, 5, 20, 30, 50. As expected, the performance improved with increasing value of α. A larger α value tried to maintain the overlap avoiding oscillations.

Even though simulated annealing is a very powerful technique, it has the uncertainties associated with a randomized approach. As a result, it is best suited for performing a number



(a)

(b)

Fig. 15. Results for reconfiguration involving (a) serial configuration and (b) loop configuration using $E(5)$ for different α values.

of off line simulations and then using the best one out of those to reconfigure the robot instead of real time application.

VIII. Conclusions

In this paper we define metrics which measure distance between configurations of a metamorphic system. We then illustrate how these metrics are applied to the motion planning/self-reconfiguration of metamorphic robotic systems. The method of simulated annealing was used with these metrics as the energy function for a variety of initial and final configurations (both simply connected and configurations containing loops). It was shown that the perfomance of simulated annealing using the metrics developed in this paper performs better than with other cost functions.

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.

References

[1] H. Asama, K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, and I. Endo, "Collision avoidance among multiple mobile robots based on rules and communication," in *IROS'91*, pp. 1215–1220.

[2] B. Benhabib, G. Zak, and M. G. Lipton, "A generalized kinematic modeling method for modular robots," *J. Robot. Syst.*, vol. 6, no. 5, 1989, pp. 545–571.

[3] G. Beni, "Concept of cellular robotic systems," in *IEEE Int. Symp. Intell. Contr.*, Arlington, VA, Aug. 24–26, 1988.

[4] G. Beni and J. Wang, "Theoretical problems for the realization of distributed robotic systems," in *1991 IEEE Conf. Robot. Automat.*, pp. 1914–1920.

[5] V. Bryant, *Metric Spaces: Iteration and Application*. New York: Cambridge Univ. Press, 1985.

[6] I.-M. Chen and J. W. Burdick, "Enumerating non-isomorphic assembly configurations of a modular robotic system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Yokohama, Japan, July 1993, pp 1985-1992.

[7] G. S. Chirikjian, "Kinematics of a metamorphic robotic system," in *Proc. 1994 IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 1994, pp. 449–455.

[8] G. S. Chirikjian, A. Pamecha, and I. Ebert-Uphoff, "Evaluating efficiency of self reconfiguration in a class of modular robots," *J. Robot. Syst.*, June 1996.

[9] R. Cohen, M. G. Lipton, M. Q. Dai, B. Benhabib, "Conceptual design of a modular robot," *J. Mech. Design*, pp. 117–125, Mar. 1992.

[10] E. T. Copson, *Metric Spaces*. Cambridge, England: Cambridge Univ. Press, 1968.

[11] J. R. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs*. New York: Marcel Dekker, 1992.

[12] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic systems," *Proc. 1988 IEEE Conf. Robot. Automat.*, pp. 1581–1586.

[13] T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," in *Proc. 1990 IEEE Conf. Robot. Automat.*, pp. 662–667.

[14] T. Fukuda, Y. Kawauchi, and F. Hara, "Dynamic distributed knowledge system in self-organizing robotic systems; CEBOT," in *1991 IEEE Conf. Robot. Automat.*, 1991, pp. 1908–1913.

[15] H. Gericke, *Lattice Theory*. New York: Frederick Ungar, 1966

[16] S. Hackwood and J. Wang, "The engineering of cellular robotic systems," *IEEE Int. Symp. Intell. Contr.*, Arlington, VA, Aug. 24-26, 1988.

[17] S. Hackwood and G. Beni, "Self-organizing sensors by deterministic annealing," in *IROS'91*, Osaka, Japan, pp. 1177–1183.

[18] ——, "Self-organization of sensors for swarm intelligence," *1992 IEEE Conf. Robot. Automat.*, pp. 819–829.

[19] L. Kelmar and P. K. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," in *Proc. 1988 IEEE Conf. Robot. Automat.*, pp. 663–668.

[20] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, no. 4598, pp. 671–679, May 13, 1983.

[21] D. Kőnig, "Graphen und matrizen," *Mat. Fiz. Lapok.*, vol. 38, pp. 116–119, 1931.

[22] H. W. Kuhn, "The Hungarian methods for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, pp. 83–97, 1955.

[23] J. E. Marsden, *Elementary Classical Analysis*. New York: W. H. Freeman, 1974.

[24] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, *J. Chem. Phys.*, vol. 21, pp 1087-1092, 1953.

[25] S. Murata, H. Kurokawa, S. Kokaji, "Self-assembling machine," in *Proc. 1994 IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 1994, pp. 441–448.

[26] ——, "Self-organizing machine," in *Video Proc., 1995 IEEE Int. Conf. Robot. Automat.*, Nagoya, Japan, May 1995.

[27] A. Pamecha and G. S. Chirikjian, "A metamorphic robotic system: simulation and connector mechanism demonstration" in *Video Proc., 1995 IEEE Int. Conf. Robot. Automat.*, Nagoya, Japan, May 1995.

[28] A. Pamecha, Y. Kohaya, and G. S. Chirikjian, "Design of error tolerant coupling mechanisms for metamorphic robots," in *Third IASTED Conf. Robot. Manufact.*, Cancun, Mexico, June 1995, pp. 7-13.

[29] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization, Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

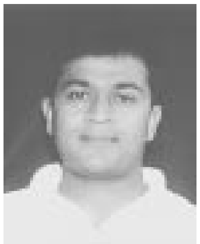
[30] F. Roberts, *Applied Combinatorics*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[31] M. Sciaky, "Modular robots implementation," in *Handbook of Industrial Robotics*, S. Nof, Ed. New York: Wiley, 1985, pp. 759–774.

[32] K. H. Wurst, "The conception and construction of a modular robot system," in *Proc. 16th Int. Symp. Indust. Robot.*, 1986, pp. 37–44.

[33] Yim, M., "A Reconfigurable Modular Robot with Many Modes of Locomotion," in *Proc. 1993 JSME Int. Conf. Advanced Mechatron.*, Tokyo, Japan, Aug. 1993, pp. 283–288.

[34] M. Yim, "New locomotion gaits," in *Proc. 1994 IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 1994.



Amit Pamecha (S'95) received the Bachelors degree in mechanical engineering from the Indian Institute of Technology, New Delhi, India, in 1993 and the Masters degree from the Johns Hopkins University, Baltimore, MD, in 1995.

He is currently working at Systems Research and Applications Corporation, Fairfax, VA.

Mr. Pamecha is a member of the IEEE Robotics and Automation Society and the IEEE Computer Society.

Imme Ebert-Uphoff received the degree Diplom der Techno-Mathematik in 1993 from the University of Karlsruhe, Germany, and the Ph.D. degree in 1997 from the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD. Her thesis research focused on the development of discretely actuated, hyper-redundant manipulators with an emphasis on workspace generation and inverse kinematics.

She is currently a post-Doctoral researcher at Laval University, Canada. Other research interests include the study of parallel platform mechanisms and the development of simulation tools for that purpose.

Gregory S. Chirikjian (M'93) received the B.S.E. degree in engineering mechanics and the M.S.E. degree in mechanical engineering in 1988 while also fulfilling requirements for the B.A. degree in mathematics, all from Johns Hopkins University, Baltimore, MD.

Between 1988 and 1992 he was a graduate student at the California Institute of Technology, Pasadena, where he did his Ph.D. work under the supervision of Prof. J. Burdick. Since 1992, he has been with the Department of Mechanical Engineering, Johns Hopkins University, where he is now an Associate Professor. His research interests include the kinematic analysis, motion planning, design, and implementation of "hyper-redundant," "metamorphic," and "binary" manipulators.