Multi-robot Formation Morphing through a Graph Matching Problem

Lantao Liu and Dylan A. Shell

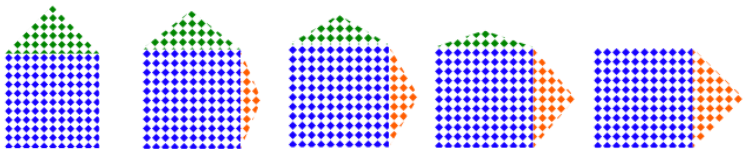
Abstract We consider the problem of changing smoothly between formations of spatially deployed multi-robot systems. The algorithm presented in this paper addresses scenarios in which gradual and seamless formation transitions are needed, a problem which we term *formation morphing*. We show that this can be achieved by routing agents on a Euclidean graph that corresponds to paths computed on — and projected from— an underlying three-dimensional matching graph. The three-dimensional matching graph is advantageous in that it simultaneously represents a logical assignment problem (for which an optimal solution must be sought) and metric information that comprises the spatial aspects of the Euclidean graph. Together, these features allow one to find concurrent disjoint routing paths for multiple source multiple goal (MSMG) routing problems, for which we prove one may find routing solutions to optimize different criteria. These disjoint MSMG paths efficiently steer the agents from the source positions to the goal positions, the process of which enables the seamless transition from an old formation to a new one.

1 Introduction

Part of multi-robot formation control involves manoeuvring a spatially dispersed system from one formation to another. Formation control has received a great deal of attention and extensive investigation in the past decades (see reviews of Murray; Chen and Wang [2007; 2005]). Most previous studies consider formation control of the whole system, but, in many situations, only parts of the system need to be changed to reach a new formation. For example, sometimes only patches of agents in certain corners need move to other locations, or boundary agents need to fill inner holes. If the majority the system keeps its structure unchanged, while a minority migrate to other places, then an incremental variation of the problem is worth addressing. We call formation control in such scenarios *formation morphing* since the formation is changed as if it is gradually "deformed" in places, while the major pattern is unaltered. Fig. 1 shows an example of seamless formation morphing.

Recently we have shown that by exploring the matching graph version of the Hungarian method [Kuhn, 1955] and interpreting it in three dimensional space, the

Lantao Liu, Dylan A. Shell
Dept. of Computer Science and Engineering, Texas A&M University, College Station, TX, USA
e-mail: {lantao, dshell}@cse.tamu.edu



(a) (b) (c) (d) (e)

Fig. 1 Evolution of formation morphing. Source nodes are colored in green (top triangles) and gradually disappear. Goal nodes are colored in orange (right triangles) and gradually grow.

assignment problem can also be used to deal with the standard routing problems [Liu and Shell, 2012]. In this paper we focus on more complicated conditions involving multiple paths generated from the matching graph, and develop an assignment-based formation control method for multi-robot systems for these cases. Specifically, formation morphing is achieved by routing certain agents from their source/initial positions to the defined goal positions along some trajectories, such that all agents involved in the trajectories simultaneously shift to and thus replace their successive neighbors. This is a process which gradually morphs the formation into a new shape. The routing trajectories are a set of interference-free paths on the Euclidean graph in which the nodes are the agents and edges are the traversal links; the paths are projected from the Hungarian augmenting paths in the 3D bipartite graph which we construct. One important contribution of this work is that the formation morphing is carried out in ways which optimize useful criteria, *e.g.*, the overall travel cost is minimized, the total interruptions are minimized (the number of the robots re-deployed is fewest), and the number of disjoint paths that are allowed is maximized. This is because the routing is incorporated in, and projected from, the matching graph from which globally optimal solutions of assignment problems are sought and found.

More specifically, the contributions of this work include:

- The design of a formation control strategy through routing paths projected from a 3D matching graph, which combines the logical description of the matching graph and the spatial embedding of the Euclidean graph.
- An optimal means for producing routing solutions of interest in applications; for example, the global minimal travel distance and shortest hopping distance are analyzed.
- Simultaneous generation of disjoint and conflict-free MSMG paths. Conditions for producing disjoint paths, and the maximal number of such paths are analyzed.

2 Related Work

Formation control is an active topic in the multi-robot research area and many approaches for controlling the formations of various types have emerged during the past decades. To sample from as many distinct taxonomical branches as possible, we acknowledge control theoretic schemes [Fax and Murray, 2004; Hsieh *et al.*, 2008], strategies with combinatorial optimizations [Michael *et al.*, 2008], approaches dealing with geometry and potential fields [Consolini *et al.*, 2007; Song and Kumar, 2002], as well as behavior-based methods [Balch and Arkin, 1997] and methods employing biological inspired mechanisms [Reynolds, 1987;

Tanner *et al.*, 2007]. We are particularly interested in the formations of structured and well-aligned patterns, in which agents keep similar distances from each other (this somewhat mimics certain animal behaviors, *e.g.*, schools of fish, cattle herds, inserts swarms, *etc*.). Works dealing with the formations (patterns) that are similar to this work include [Alonso-Mora *et al.*, 2011; Kurabayashi *et al.*, 2009; Ravichandran *et al.*, 2007], although their methods differ significantly.

One may also regard formation control as the assignment of robots to goal positions that define the final pattern. Smooth shifting of formation shapes addressed by this paper relates directly to the reassignment of robots to tasks (work specifically addressing reallocation includes that of Karmani *et al.* [2007] and Shen and Salemi [2002]), and to controllers which enforce some metric (or shape) constraints (notable recent examples of formation control include that of Michael *et al.* [2008], Liu *et al.* [2008], and Ren and Sorensen [2008]). A clear, recent example of reallocation and formation work together is that of Agmon *et al.* [2010]. The authors designed a polynomial time graph-based method to extract a subset of the robots from a coordinated group so that this subset can perform a new task while minimizing the cost of interacting with the remaining group.

This paper offers a different perspective: spatial formation transitions of a multi-robot team are achieved by routing agents on the 2D Euclidean graph but doing this by regarding it as a projection from the 3D representation of a corresponding matching graph. The approach has been described in detailed in our recent work [Liu and Shell, 2012] where we focused on the analysis of single path properties. In this work we emphasize multi-path conditions which are more complicated and reveal the merits of thinking about formation transitions in this way. The underlying method is the same incremental matching approach, *viz.* execution of stages of the Hungarian Method which produce paths with desired global optimization properties by incorporating both (metric) traversal information and reallocation (logical) costs, simultaneously.

3 Synthesized Matching Graph and Assignment Problem

This work is based on two forms of graphs: the Euclidean graph and the bipartite graph (or *bigraph* for short).

The Euclidean graph is a standard graph $G = (V,E)$ with a metric embedding so that the vertices in V describe locations and edges in E express distances between the vertex pairs. We let each vertex of G denote an agent, and let $w(i,j) = -d(i,j)$ represent the weight of edge $e(i,j) \in E$, where $d(i,j)$ is the travel distance between agent pair (i,j). The negated travel costs transform the problem from minimization to maximization. This transformation does not change the optimization objective but makes the problem consistent with the assignment utility maximization described below. In addition, traversability constraints, limited sensing/communication ranges, and so one, imply that the graph G is likely to be sparse.

The Bigraph is the main data structure used in the Hungarian algorithm [Kuhn, 1955]. In the Hungarian algorithm (see Algorithm 1), is one of the most well-known optimal assignment algorithms and can efficiently solve an $n \times n$ assignment prob-

Algorithm 1 The Hungarian Algorithm

Require:

An $n \times n$ assignment matrix represented as the complete weighted bigraph $\tilde{G} = (X, Y, \tilde{E})$, where $|X| = |Y| = n$.

Ensure:

A perfect matching M.

1: Generate initial labellings $l(\cdot)$, and an initial matching M in G_e.
2: If M perfect, terminate algorithm. Otherwise, randomly pick an exposed vertex $u \in X$. Set $S = \{u\}$, $T = \varnothing$.
3: If $N(S) = T$, update labels:
$\delta = min_{\forall x \in S, y \in Y \setminus T} \{l(x) + l(y) - \tilde{w}(x,y)\}$

$$l'(v) = \begin{cases} l(v) - \delta & \text{if } v \in S, \\ l(v) + \delta & \text{if } v \in T, \\ l(v) & \text{otherwise.} \end{cases}$$

4: If $N(S) \neq T$, pick $y \in N(S) \setminus T$.
(a) If y exposed, then $u \rightsquigarrow y$ is an augmenting path, then augment matching M and go to step 2.
(b) If y matched, say to z, extend the tree: $S = S \bigcup \{z\}, \quad T = T \bigcup \{y\}$, and go to step 3.

Notes & Definitions:

- $\tilde{w}(x,y)$ is the weight of edge $\tilde{e}(x,y)$.
- Equality graph $G_e = \{\tilde{e}(x,y) : l(x) + l(y) = \tilde{w}(x,y)\}$.
- Neighbor of vertex $u \in X$: $N(u) = \{v : \tilde{e}(u,v) \in G_e\}$.

lem in $O(n^3)$ time. In the algorithm, the bigraph $\tilde{G} = (X, Y, \tilde{E})$ is another representation of utility matrix $U = (u_{ij})_{n \times n}$, where X and Y respectively denote the set of agents and tasks, and the set $\tilde{E} = \{\tilde{e}(i,j)\}$ are edges weighted by the utilities ($\tilde{w}(i,j) = u_{ij} = -d(i,j)$) between associated agent-task pairs (i,j), $i, j = 1, \cdots, n$. Since the bigraph represents matching relationships , sometimes it is also called the *matching graph*. The assignment problem is a matching problem where the goal is to find a set of *maximally weighted* and *mutually excluded* edges that constitute a perfect matching M such that each agent in X is uniquely assigned to a task in Y.

The Hungarian algorithm grows a matching by searching for a path, called an *augmenting path*, which consists of an alternating sequence of *matched* and *unmatched* edges but with free end nodes. This means the quantity of unmatched edges is an odd number and is exactly one more than number of the matched edges. The algorithm augments the set of matched edges by simultaneously flipping the matched and unmatched edges in the augmenting path. (Formal definitions of these operations on matchings are omitted, refer to Lovász and Plummer [1986].) In Algorithm 1, steps 2 to 4 describe the procedure of seeking and flipping an augmenting path. We call a single iteration of this procedure a *stage* (see Fig. 2). Note that each stage finds exactly one augmenting path which increases the size of the matching by exactly one. Thus, the algorithm requires at most n stages to obtain all n matched edges with mutually excluded end nodes, thereby forming the optimal assignment solution.

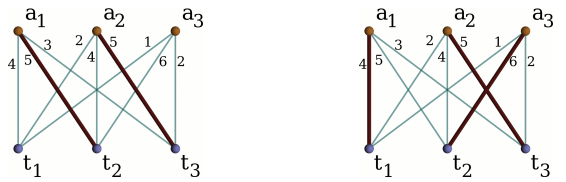


Fig. 2 (a) Two matched edges found after running two stages of the algorithm; (b) A perfect matching consisting of three matched edges is found after one additional stage (by augmenting path $a_3 \to t_2 \to a_1 \to t_1$).

In its conventional use of finding a set of matchings, the bigraph $\tilde{G} = (X, Y, \tilde{E})$ is only interpreted as having meaning in a logical sense; any geometric information used to form the utilities is typically ignored. This differs from the embedded Euclidean graph $G = (V, E)$ which encodes the spatial description directly. But being graphs, both forms have common characteristics, for instance, both the Euclidean graph and the bigraph can be represented with matrices: G can be represented with a symmetric adjacency matrix with $w(i, j)$ as entries, and $\tilde{G}$ can be represented with a non-symmetric utility matrix with $\tilde{w}(i, j)$ as entries. This suggests that perhaps the bigraph may have the potential to express spatial information adequately *if the utility matrix is symmetric*.

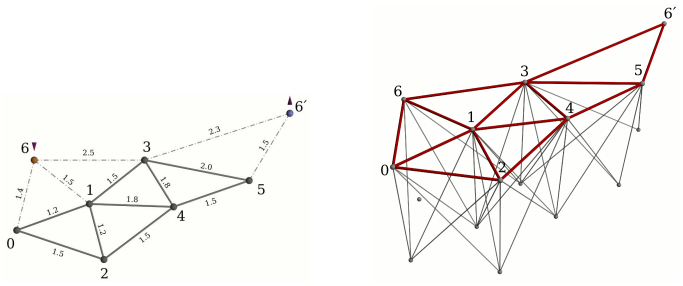
All off-diagonal entries of the two matrices have the following relationship:

$$\begin{aligned} \tilde{w}(i, j) = \tilde{w}(j, i) = w(i, j) = w(j, i), \\ \forall i \neq j, 1 \leq i, j \leq n. \end{aligned} \tag{1}$$

This means that if we ignore the diagonal entries, the assignment utility matrix is the form of an adjacency matrix, which is *the basic idea in bringing the two forms of graph together in the construction of a unified one*.

This synthesized graph may be imagined as if an identical copy of the Euclidean graph G had been lifted and placed over G. Via this "extrusion" a three dimensional mesh is formed with two identical layers plus all edges that connect them. Here the two layers correspond to the two partitions of a bigraph, *i.e.*, the bigraph vertex sets satisfy $X = Y = V$. Each edge $e(i, j) \in G$ is replaced with a pair of edges $\tilde{e}(i, j) \in \tilde{G}$ and $\tilde{e}(j, i) \in \tilde{G}$. (Note: different from edges in G, in $\tilde{G}$ edges $\tilde{e}(i, j) \neq \tilde{e}(j, i)$ since i, j are nodes from different vertex sets—either X or Y.) An example is illustrated in Fig. 3(b). A more detailed description of this transformation is provided in our preceding work Liu and Shell [2012]. In the remainder of the paper we assume that the vertices X in the top layer represent the agent set and vertices Y in the bottom layer denote the task set. Since nodes of either layer are copies from the Euclidean graph, this synthesized graph thus also conveys information about the spatial locations (top nodes describe the agent locations, and bottom nodes describe the task locations). If an agent node is matched to a task node, the agent needs to move from its current location to the newly assigned task location, and when a pair of agent and task nodes are vertically aligned, one can simply imagine that the agent has reached its deployed location and completed the position shift, so need not relocate.

Because this synthesized graph is a matching graph, we call it the *3D bigraph* (*3D matching graph*) and continue to denote it with symbol $\tilde{G}$. Thus, we obtain a mapping (projection) $\Omega : G \to \tilde{G}$. With known $G = (V, E)$, the projection $\tilde{G} = \Omega(G) = (X, Y, \tilde{E})$. To get $G = \Omega^{-1}(\tilde{G})$, an inverse operation is carried out, analogously.



(a) (b)

Fig. 3 Mapping from Euclidean graph to bigraph. (a) An Euclidean graph of networked robots with only nearest neighbors connected. This example illustrates the simple case of morphing one agent: vertices 6 and 6′ denote the initial and goal nodes/positions, respectively; (b) The corresponding sparse bigraph visualized in 3D. Bold red edges on top layer do not exist but just show the projection relationship with graph on the left.

4 Morphing the Formation

Formation morphing is done by seamlessly transferring agents from the source positions to the predefined goal positions as illustrated in Fig. 1. It can be imagined as cutting a batch of nodes from the source regions and pasting them into the goal regions. Naturally this implies that the two regions must be determined before the morphing operation is begun. Assume that agents in the source positions form a set A, and nodes in the goal positions form set B. Note that a goal location has no agent in it but will be occupied by an agent after the morphing process is finished. For each agent node in A, if it is connected with a unique goal node in B by a routing path, and if we let all agent nodes on this path shift to their successors' locations in a chain, then it looks as if this agent in A is sent to B, and all other nodes in $V \setminus A$ are still occupied by unique agents. This is also the essence of new agent-task insertions to the existing assignment described in our preceding work [Liu and Shell, 2012]. Different from that, formation morphing requires multiple interference-free paths to simultaneously steer multiple agents from the source positions to unique goal positions, which is the *multiple sources multiple goals* (MSMG) routing problem.

In this paper, we provide a solid analysis for the generated MSMG paths, as well as the details of applying our method to control formation transitions. We start the analysis by assuming that each node of A and B is directly traversable (with at least one edge connecting) to some nodes in $V \setminus A$. The MSMG routing paths are obtained by projecting the *matched edges* of augmenting paths in 3D bigraph to either planar layer. In order to get these augmenting paths connecting A and B from the Hungarian algorithm, these rules need be followed to construct the 3D bigraph: the top layer is split into two subsets of nodes— A and $X \setminus A$, and the bottom layer is also separated

into two subsets— B and $Y \setminus B$. Edges are added between the two layers following the 3D bigraph construction procedure described in Section 3. Nodes of $X \setminus A$ and $Y \setminus B$ are vertically aligned and initialized as matched to represent the stationary intermediate nodes, whereas all other edges are initialized as unmatched. There are only $|A|$ nodes unmatched/un-assigned (assuming $|A| \leq |B|$), and each stage of Hungarian algorithm costs $O(|V|^2)$ time complexity, therefore only $O(|A||V|^2)$ is required to compute all MSMG paths.

4.1 Interference-free Property of MSMG Paths

Let $P : A \rightsquigarrow B$ denote the set of paths that connect nodes in A and nodes in B. Routing paths P and Q are *disjoint* paths if no node or edge is shared between them. A shared node means that the agent at the intersection of different paths is required to simultaneously replace multiple other agents on corresponding paths, which is impossible.

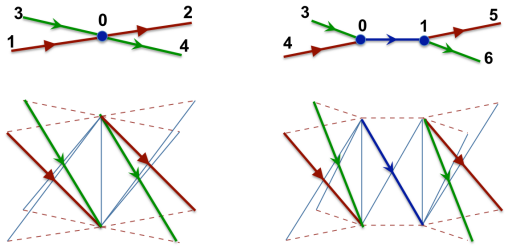


Fig. 4 Neither node nor edge will be shared among multiple paths produced from a matching graph. (a) Assumption of a shared node 0 at the crossing of path $1 \to 0 \to 2$ and path $3 \to 0 \to 4$. The bottom graph is the 3D bigraph showing the violation of mutual exclusion constraint; (b) Assumption of a shared edge $e(0,1)$ belonging to both path $3 \to 0 \to 1 \to 6$ and path $4 \to 0 \to 1 \to 5$.

Theorem 1. *The Hungarian algorithm run on bigraph $\tilde{G} = \Omega(G)$ produces only disjoint paths on graph G.*

Proof. We prove this theorem via contradiction: if the generated paths are not disjoint then there must exist at least two paths with a shared node that crosses between them. An example is illustrated in Fig. 4(a). Since P is also comprised of vertices and edges, it can be denoted by $P = (V^p, E^p)$, alternatively. Assume m paths $P_i = (V_i^p, E_i^p) \subseteq G$ ($m > 1$ and $i = 1, 2, \cdots, m$) are generated from the Hungarian algorithm, and there is shared (crossing) node $v_s \in \bigcup_{\forall i} P_i$ having more than one incoming routing edge and more than one outgoing routing edge. More than two routing nodes must be connected to v_s:

$$|\{u \mid e(v_s, u) \in P_i, \forall i = 1, 2, \cdots, m\}| > 2. \tag{2}$$

This means that in bigraph $\tilde{G} = \Omega(G)$ either the corresponding agent node (in top layer of Fig. 4(a)) or the task node (in bottom layer) or both have more than one matched edge, which contradicts the mutual exclusion constraint and violates the

feasibility of the assignment solution.

The shared edge case is analogous. □

4.2 Optimality Analysis of MSMG Paths

Thus far we have not described how the diagonal entries in the utility matrix are determined during the construction of a 3D bigraph. The diagonal values are actually the weights of the vertical edges from agents in set $V \setminus A$, see Fig. 3(b) for an example. Producing utility matrices with different diagonal weights, and feeding them to the Hungarian algorithm will produce distinct MSMG paths. Here we show two conditions that yield optimizations of particular interest.

Theorem 2. *The set of MSMG routing paths $P : A \rightsquigarrow B$ projected onto the Euclidean graph from the matching computed with the Hungarian algorithm will minimize the global hopping distance*[1] *$\mathscr{D}(P)$ when the weights $\tilde{w}(i,i)$ $(\forall i \in V \setminus A)$ (diagonal utilities) are sufficiently large.*

Proof. Let ζ be the largest absolute value of the utility matrix, *i.e.*,

$$\zeta = \max\{-\min(U), \max(U)\}, \tag{3}$$

and let $\pi = |V|\zeta + \varepsilon$, where ε is a small positive value. Weights of the edges $\tilde{e}(i,i)$ can be made sufficiently large by letting $\tilde{w}(i,i) = \pi$ for all nodes in $V \setminus A$. Now assume there exists another path $Q : A \rightsquigarrow B$ with a shorter hopping distance $\mathscr{D}(Q) < \mathscr{D}(P)$. Since all nodes not on the paths themselves maintain their matching, the weight sums $f_s(\cdot)$ for the two matching solutions are

$$f_s(P) = \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) + (|V| - \mathscr{D}(P))\pi, \tag{4}$$

and

$$f_s(Q) = \sum_{\forall_{(i,j)} e(i,j) \in Q} \tilde{w}(i,j) + (|V| - \mathscr{D}(Q))\pi, \tag{5}$$

respectively. Since

$$\begin{aligned} f_s(P) - f_s(Q) &= \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) - \sum_{\forall_{(i,j)} e(i,j) \in Q} \tilde{w}(i,j) + (\mathscr{D}(Q) - \mathscr{D}(P))\pi \\ &\leq \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) - \sum_{\forall_{(i,j)} e(i,j) \in Q} \tilde{w}(i,j) - \pi \\ &\leq \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) - \pi < 0, \end{aligned} \tag{6}$$

contradicting the optimality of the matching from the Hungarian algorithm. □

Theorem 3. *When $w(i,i) = 0$ $(\forall i \in V \setminus A)$, the set of MSMG routing paths $P : A \rightsquigarrow B$ computed by projecting the Hungarian algorithm's perfect matching to the Euclidean graph have the globally shortest path length.*

[1] Hopping distance is also called Geodesic distance, it is the quantity of edges in the path and therefore measures the number of nodes involved and interrupted in the deployment.

Proof. When $\tilde{w}(i,i) = 0, \forall i \in V \setminus A$, the weight sum of the matching solution for the assignment problem is

$$\begin{aligned} f_s(P) &= \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) + \sum_{\forall v \notin P} \tilde{w}(v,v) \\ &= \sum_{\forall_{(i,j)} e(i,j) \in P} \tilde{w}(i,j) + 0 = \sum_{\forall_{(i,j)} e(i,j) \in P} w(i,j), \end{aligned} \tag{7}$$

which is essentially the total length of all MSMG routing paths. □

These two path properties are important since the globally shortest paths minimizes the total travel distances for a morphing operation, whereas the globally shortest hopping distance represents the fewest interruptions to the system (an interruption usually bears a cost).

4.3 Concurrent Paths in Narrow Bridges

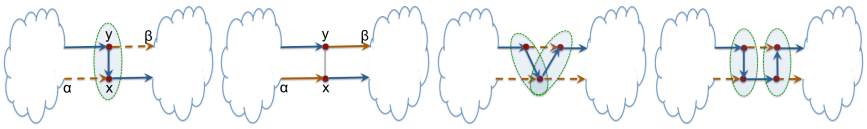
In investigating paths formed in complex environments, it is important to quantify the maximum number of concurrent paths that can be formed. Narrow spaces may pose a challenge because they can impose a limit on the degree of concurrency that is possible; understanding these limits allows one to decide when sequential treatment (*e.g.,* for subsets of A and B) might be called for.

Definition 1. Let $G = (V,E)$ be a connected graph. A subset $C \subseteq V$ is called a *vertex cut* if $G \setminus C$ (the remaining graph after removing all vertices in C and their incident edges) is disconnected. A *minimal vertex cut* is a vertex cut with the least cardinality.

Definition 2. The *local connectivity* $\kappa(u,v)$ (or $\kappa(A,B)$) is the size of a smallest vertex cut separating non-adjacent vertices u and v (or vertex sets $A \subseteq V$ and $B \subseteq V$).

Theorem 4. *(*Menger's Theorem*) Let $G = (V,E)$ be a graph and $A,B \subseteq V$, then $\kappa(A,B)$ is equal to the maximum quantity of disjoint A-B-paths (*i.e.*, the paths that connect vertices of A and B) in G.*

Proof. Three proofs appear in Diestel [2005].



(a) (b) (c) (d)

Fig. 5 Conditions on concurrent paths passing through a narrow bridge in the graph. (a) Nodes x and y on a path (solid arrowed edges) are from the same minimal vertex cut circled in an ellipse; (b) Two paths are generated after having found an augmenting path $\alpha \to x \to y \to \beta$; (c)-(d) Vertices of a path are from intersecting and independent minimal vertex cuts, respectively.

Theorem 4 provides the upper bound for the quantity of possible disjoint routing paths. However, we wish to know how close to this bound the disjoint paths produced by the Hungarian algorithm on the corresponding matching graphs are.

Theorem 5. *For connected graph $G = (V,E)$ with $A,B \subseteq V$ and $\min\{|A|,|B|\} \geq \kappa(A,B)$, the number of disjoint paths generated from Hungarian algorithm is equal to $\kappa(A,B)$,* i.e.*, Hungarian algorithm running on $\tilde{G} = \Omega(G)$ outputs a set of disjoint A-B-paths, such that each path consists of exactly one cut vertex belonging to a minimal set of cut vertices.*

Proof. Assume a maximal set of disjoint paths $S_p = \{P_i\}$, $i = 1,\cdots,m$ is output, and assume a minimal vertex cut of G is C. If $|S_p| < |C|$, there must be some path P_l ($l \in [1,m]$) that contains more than one cut vertex from C. Assume these vertices form a set $V_l^{'} \subseteq V$ with $|V_l^{'}| = K$, then there must be $K-1$ edges $E_l^{'} \subseteq E$ (which can also be path segments) connecting these vertices. For an arbitrary edge $e(x,y) \in E_l^{'}$ where $x,y \in V_l^{'}$, x,y must be incident with other edges that are not on routing paths (a property following from the mutual exclusion constraint and the definition of a minimal cut), assume they are $e(\alpha,x)$, $e(y,\beta)$ respectively (illustrated in Fig. 5(a)). Then path $e(\alpha,x)$, $e(x,y)$, $e(y,\beta)$ forms an augmenting path and flipping of matched and unmatched edges cancels edge $e(x,y)$ to effectively bridge two new paths (example shown in Fig. 5(b)). Similarly, other edges in $E_l^{'}$ can also be revised and cancelled, and each such revision will add exactly one new path. Other complex conditions involving multiple intersecting or independent sets of minimal vertex cuts (see. Fig. 5(c) and 5(d)) are treated analogously. There must be $\kappa(A,B)$ disjoint paths generated, each of which consists of exactly one vertex from a minimal vertex cut. □

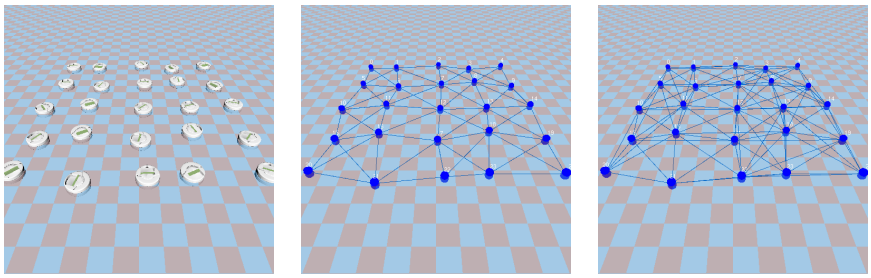
4.4 Morphing with Multiple Shifts

In complex scenarios, we cannot directly generate all MSMG paths in one go. This includes the condition of $\kappa(A,B) < \min\{|A|,|B|\}$, in which at most $\kappa(A,B)$ paths can be generated at one time. Another condition occurs when some nodes in A are not directly traversable to nodes in $V \setminus A$ (no edges directly connect them), these nodes must first morph (move) to some locations near enough to bridge to nodes of $V \setminus A$. In these cases, paths can only be produced in multiple batches and agents may need to carry out several shifts to complete the morphing task.

Generally, those peripheral agents need to be "transferred" first since otherwise inner holes may exist, which also deteriorates the traversability. (Peripheral nodes can be detected in a decentralized way as described in Liu *et al.* [2011].) If local connectivity does not allow all peripheral nodes to morph at one time (via conditions given above), then the remaining peripheral nodes are queued and will be processed with high priority in next iteration to first get routing paths. An update of nodes' positions exposes new peripheral agents which can be processed in the following iterations. All future position shifts of an agent are ordered since waypoints require the agent complete movement one at a time. This is exactly the process of formation morphing.

5 Results

We simulated the algorithm with tens to hundreds of robots in order to validate the presented method. The simulator is written in C++ and it runs in a GNU/Linux environment. We assume that all robots are homogeneous and have identical sensing and communication ranges, within which each robot is capable of recognizing its neighbors as well as their distances and bearings. Spatial constraints and sensing/communication ranges may limit each agent's traversablility to only its nearest neighbors. In this work, a traversal link (edge) is added if the distance between a pair of agents is less than a given threshold. As a result, differing thresholds on traversal link length form Euclidean graphs of different sparsity, as illustrated in Fig. 6. To permit easy visualization, robots are simplified as dots, and the formation of the system is aligned in arrays to better reveal the concurrent paths and to show the shifting motions during the morphing. Also, the robots are assumed to autonomously avoid obstacles locally to avoid potential collisions (*e.g.*, making a minimal detour around a obstacle if need be). The algorithm applies to any formation structures so long as the underlying Euclidean graph is connected.



(a) (b) (c)

Fig. 6 (a) A formation of robots; (b)–(c) Euclidean graphs with traversal edges of different lengths.

Fig. 7 shows the evolution of an instance of formation morphing. In Fig. 7(b), agents form a triangle pattern, and the goal is to transform into the same size triangle rotated by 180°. Nodes in source positions are identified with green solid dots (forming set A), and the nodes in goal positions are orange circles (forming set B). In this example all nodes in A can traverse directly to nodes in $V \setminus A$.

Fig. 7(b)—7(d) and Fig. 7(e)—7(h) show two morphing processes (the difference lies in different traversal edge lengths, the former have larger thresholds than the latter). Optimization of global hopping distance and travel distances yield the same routing solutions in this example since the formation structure is well-aligned and the traversal link distance threshold is uniform. Since all agents involved carry out the relocation simultaneously, the formation transitions are achieved efficiently.

Fig. 8 illustrates the multiple shifts discussed in Section 4.4. Fig. 8(a) shows the initial and goal formations, and the task is to morph the top and bottom extruded agents to the left and right sides. Unlike the previous example, the traversal edge lengths are short so that the outer layer of green (source) nodes are not directly connected to the inner blue (intermediate) nodes. Two processes with differing thresh-

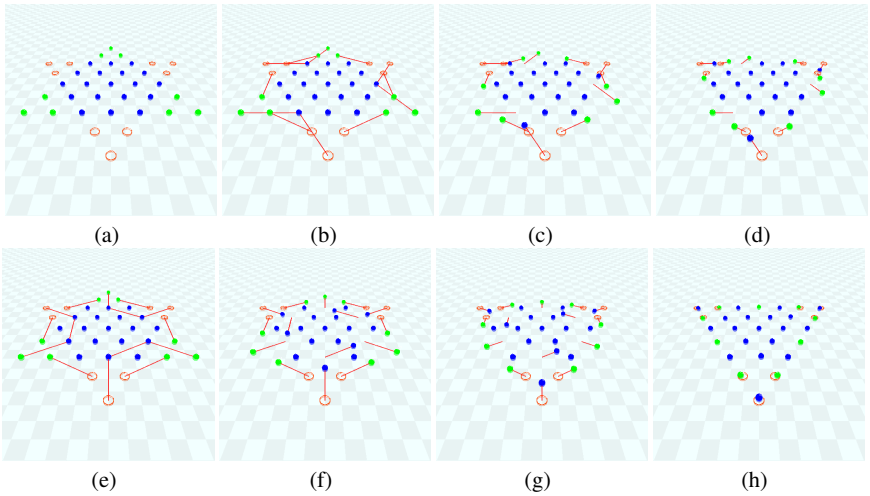


Fig. 7 (a) Initial and goal formations; (b)—(h) Processes of formation morphing.

olds are provided in Fig. 8(a)—8(d) and Fig. 8(e)—8(h), respectively. In particular, Figs. 8(d) and 8(g) show the second round of morphing paths.

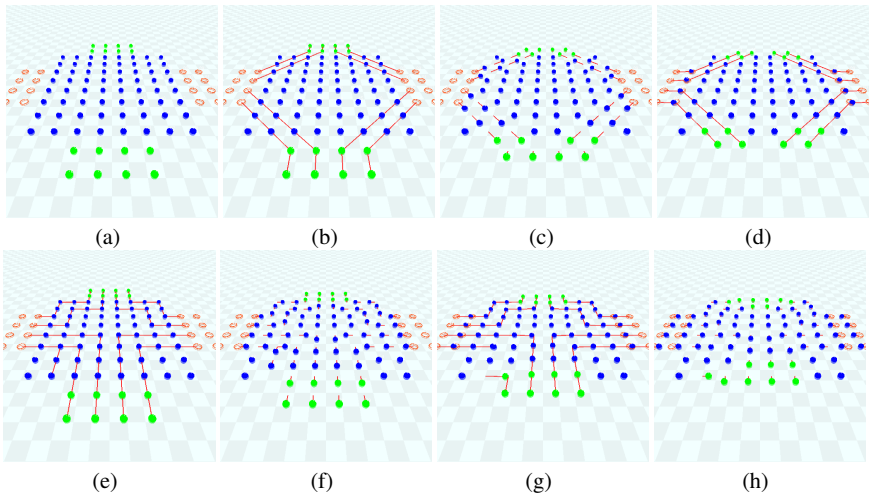
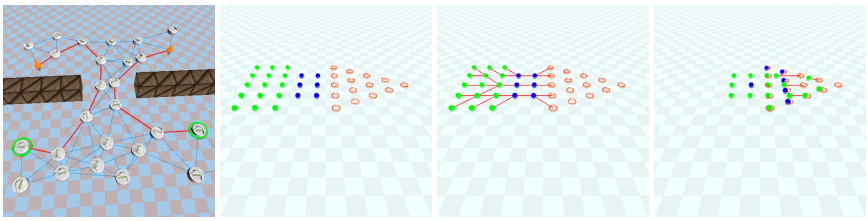


Fig. 8 (a) Initial and goal formations; (b)–(h) Processes of formation morphing.

Next, we examine the conditions for morphing paths across regions that limit the amount of concurrency. Fig. 9(a) shows a group of agents passing through a small gap. The maximal number of paths that can cross from one side to the other is two; each path passes through exactly one cut vertex in the Euclidean graph. Another example is Fig. 9(b)—9(d) where a square is morphed to a triangle while crossing a waist (one can imagine the waist part is located in between two attenuating walls

as in Fig. 9(a), and the agents need to adjust their formations while navigating the confined environment).



(a) (b) (c) (d)

Fig. 9 (a) Morphing paths across a small gap; (b)–(d) Formation morphing through a narrow bridge.

6 Discussion and Conclusion

If one compares MSMG paths generated all at once with multi-batch paths, the former have shorter global hopping/travel distances; naturally the single-batch MSMG paths fail to consider the constraints imposed by limited local connectivity or isolated nodes.In other words, these special conditions either exceed the maximal paths capacity or violate the computation rule of the Hungarian algorithm, as discussed in Section 4.4. Nevertheless, optimality of the multi-batch paths up to the constraints has not been proven. We examined these conditions and ran experiments with $\sim$50 agents, and the results show that our solutions are very close to the global optima. (Global optima are obtained by enumerating all possible cases, and on average the difference between the two solutions over the global optimum $\leq\sim 10\%$).

Another element worthy of mention is the decentralization of this algorithm. We proposed our method by assuming that a central controller has the knowledge of all agents' position information and is responsible for computing the solutions and broadcasting the results to its teammates. In distributed multi-robot systems, communication ranges are likely to be limited, which requires information to be propagated using existing multi-hop methods. Additionally, not all agents need to be involved and communicated with. Morphing can occur in a local area and 3D bigraph can be constructed with only subset of nodes if the number of source nodes and goal nodes are small and their locations are close to each other (their morphing paths will involve small number/region of nodes in $V \setminus A$).

To conclude, this paper proposes a new formation morphing strategy by simultaneously routing agents along a set of MSMG paths. Routing paths are projected from augmenting paths in a synthesized 3D bipartite graph, which combines the logical description of the matching graph and the spatial embedding of the Euclidean graph. Since the paths are computed from the optimal assignment algorithm, useful optimized properties on the paths are revealed, including the characteristic of disjointedness, the global optimality of hopping/travel distances for all same-batch paths, and the maximal number of paths given connectivity constraints. We provided different formation morphing scenarios in simulation to illustrate and validate the various distinct conditions identified in the theoretical analysis.

References

[2010] Noa Agmon, Gal A Kaminka, Sarit Kraus, and Meytal Traub. Task Reallocation in Multi-Robot Formations. *J. of Physical Agents 4(2)*, 2010.

[2011] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Roland Siegwart, and Paul Beardsley. Multi-robot system for artistic pattern formation. In *IEEE International Conference on Robotics and Automation*, pages 4512–4517, 2011.

[1997] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multi-robot teams. *Trans. on Robotics and Auto. 14(6)*, pages 926–939, 1997.

[2005] YangQuan Chen and Zhongmin Wang. Formation control: A review and a new consideration. In *IEEE/RSJ INT. CONF. INTELLIG. ROBOTS AND SYST*, pages 3181–3186, 2005.

[2007] Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. A geometric characterization of leader-follower formation control. In *2007 IEEE International Conference on Robotics and Automation*, pages 2397–2402, 2007.

[2005] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.

[2004] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *Automatic Control, IEEE Transactions on*, 49(9):1465–1476, 2004.

[2008] M. ani Hsieh, Vijay Kumar, and Luiz Chaimowicz. Decentralized controllers for shape generation with robotic swarms. *Robotica*, 26(5):691–701, 2008.

[2007] R.K. Karmani, T. Latvala, and G. Agha. On scaling multi-agent task reallocation using market-based approach. In *Intl Conf. on Self-Adaptive and Self-Organizing Systems*, 2007.

[1955] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1):83–97, 1955.

[2009] Daisuke Kurabayashi, Tatsuki Choh, Jia Cheng, and Teteuro Funato. Adaptive formation transition among a mobile robot group based on phase gradient. In *IEEE International Conference on Robotics and Biomimetics*, pages 2001–2006, 2009.

[2012] Lantao Liu and Dylan A. Shell. Tunable routing solutions for multi-robot navigation via the assignment problem: A 3d representation of the matching graph. In *Proceedings of IEEE International Conference on Robotics and Automation*, May 2012.

[2008] L. Liu, Y. Wang, S.Yang, G. Watson, and B. Ford. Experimental studies of multi-robot formation and transforming. In *Proc. of UKACC Intl. Conf. on Control*, 2008.

[2011] Lantao Liu, Benjamin Fine, Dylan A. Shell, and Andreas Klappenecker. Approximate characterization of multi-robot swarm shapes in sublinear-time. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2886–2891, 2011.

[1986] László Lovász and Michael D. Plummer. *Matching Theory*. North-Holland, 1986.

[2008] Nathan Michael, Michael M. Zavlanos, Vijay Kumar, and George J. Pappas. Distributed multi-robot task assignment and formation control. In *IEEE Intl. Conf on Robotics and Automation*, pages 128–133, 2008.

[2007] Richard M. Murray. Recent Research in Cooperative Control of Multi-Vehicle Systems. In *International Conference on Advances in Control and Optimization of Dynamical Systems*, 2007.

[2007] Ramprasad Ravichandran, Geoffrey Gordon, and Seth Copen Goldstein. A Scalable Distributed Algorithm for Shape Transformation in Multi-Robot Systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2007.

[2008] Wei Ren and Nathan Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, pages 324–333, 2008.

[1987] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, 1987.

[2002] Wei-Min Shen and B. Salemi. Distributed and dynamic task reallocation in robot organizations. In *IEEE Intl. Conf. on Robotics and Automation*, pages 1019–1024, 2002.

[2002] Peng Song and Vijay Kumar. A potential field based approach to multi-robot manipulation. In *IEEE Intl. Conf. on Robotics and Automation*, pages 1217–1222, 2002.

[2007] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in Fixed and Switching Networks. *Automatic Control, IEEE Transactions on*, 52(5):863–868, May 2007.