Vol 18 No 8, August 2009
1674-1056/2009/18(08)/3104-11

Chinese Physics B

© 2009 Chin. Phys. Soc.
and IOP Publishing Ltd

Automatic generation of min-weighted persistent formations*

Luo Xiao-Yuan(罗小元)†, Li Shao-Bao(李绍宝), and Guan Xin-Ping(关新平)

Department of Electrical Engineering, Yanshan University, Qinhuangdao 066004, *China*

(Received 13 December 2008; revised manuscript received 31 December 2008)

This paper researched into some methods for generating min-weighted rigid graphs and min-weighted persistent graphs. Rigidity and persistence are currently used in various studies on coordination and control of autonomous multi-agent formations. To minimize the communication complexity of formations and reduce energy consumption, this paper introduces the rigidity matrix and presents three algorithms for generating min-weighted rigid and min-weighted persistent graphs. First, the existence of a min-weighted rigid graph is proved by using the rigidity matrix, and algorithm 1 is presented to generate the min-weighted rigid graphs. Second, the algorithm 2 based on the rigidity matrix is presented to direct the edges of min-weighted rigid graphs to generate min-weighted persistent graphs. Third, the formations with range constraints are considered, and algorithm 3 is presented to find whether a framework can form a min-weighted persistent formation. Finally, some simulations are given to show the efficiency of our research.

Keywords: min-weighted persistent graph, rigidity matrix, minimally rigid graph, formation, multi-agent

PACC: 0210

1. Introduction

Cooperative control of multi-agent systems (MASs) is an active research area in both control theory and vehicles. Problems such as flocking, consensus, coverage and pattern formation are some of the important issues that have been studied over the past few years. Recent technological advances have spurred a broad interest in applications of MASs. Many works have propelled the development of theory and application such as the robustness of network topologies,[1] consensus protocol of MAS,[2] and micro-satellite clusters,[3,4] unmanned aerial vehicles (UAVs),[5,6] autonomous underwater vehicles (AUVs),[7,8] automated highway systems (AHSs)[9,10] and mobile robotics.[11,12]

Graph theory is widely used in formation control. In Ref.[13], Olfati-Saber presented a theoretical framework for design and analysis of distributed flocking algorithms, but his work is based on undirected graphs. In his work, one agent can communicate with all the adjacent agents, but this method will make the communications of this MAS much more complex. So, steady formation configurations with less communication complexity are being researched. During the past few years, the theories[14−19] of rigidity and persistence have established the mathematical theory for persistent formation control. Smith *et al*[20] provided some graph operations to generate minimally rigid graphs and minimally persistent graphs, and presented two algorithms based on the Pebble game and graph operations to generate minimally rigid graphs and minimally persistent graphs respectively. Hendrickx *et al*[21,22] presented some graph operations to reorganize minimally persistent formations.

The methods presented in Ref.[20]–[22] can generate the minimally persistent framework of a formation, but we will find that minimally rigid and persistent graphs are not unique. For a formation, we always want to find an economical formation configuration to minimize the communication complexity of the MAS between the agents and reduce energy consumption, which means that we need to find the min-weighted persistent framework of the formation if the edges of the framework are weighted by the distances among agents. Since the min-weighted persistent framework has the configuration of minimum communication complexity to maintain the shape of a

*Project supported by the National Natural Science Foundation for Distinguished Young Scholars of China (Grant No 60525303), the National Natural Science Foundation of China (Grant No 60704009) and Doctor Fund of Yanshan University (Grant No B203).

†Corresponding author. E-mail: xyluo@ysu.edu.cn

formation, we research into methods to find the min-weighted rigid graph and generate the min-weighted persistent formation. Fortunately, we find that the rigidity matrix is useful to search out the min-weighted rigid graph and generate the min-weighted persistent graph. Three contributions may be concluded in this paper. First, algorithm 1 is designed to generate the min-weighted rigid graph for the framework which can be represented by a complete graph; second, algorithm 2 is presented to generate the min-weighted persistent graph through directing the edges of the min-weighted rigid graph; third, the formation with range constraints is considered, and algorithm 3 is proposed to generate the min-weighted persistent formation. At the end of this paper, some simulation results demonstrate that the three algorithms are efficient.

2. Preliminaries

In this section, we will introduce some knowledge about graph theory,[23] rigid graphs,[20,22] rigidity matrix[14] and persistent graphs.[20,22]

2.1. Graph theory

A graph G consists of a vertex set $V = \{1, 2, \ldots, n\}$ and an edge set $E \subset V \times V$, where each edge $e \in E$ is a pair of vertices (i, j) such that $i \neq j$. If $(i, j) \in E$, then we say that i and j are adjacent or that j is a neighbor of i. We also define a weighted function $\delta : E \to R$ which assigns to each edge the distance between the vertices incident to that edge (i.e., $\delta(i, j) = \|p_i - p_j\|$).

If a graph G with $\forall(i, j) \in E \Rightarrow (j, i) \in E$, then we say that graph G is an undirected graph. If a graph with $\forall(i, j) \in E$ is an ordered pair of distinct vertices i, j, which is that $\forall(i, j) \in E$ does not mean $\forall(j, i) \in E$, then we say that graph G is a directed graph. Every formation can be described by an underlying graph. The vertices of the graph represent the agents in the formation and the edges represent the communications between agents. The edges of an undirected graph denote that the communications between agents are mutual while the edges of a directed graph denote that the communications are one-way.

The adjacency matrix $A = [a_{ij}]$ of a graph G is the integer matrix with rows and columns indexed by the vertices of G, such that the $(i, j) \in E$ is equal to the $a_{ij} = 1$, or else $a_{ij} = 0$. We define a weighted matrix as $W = [w_{ij}]$, where if $(i, j) \in E$, then $w_{ij} = \delta_{ij}$, or else $w_{ij} = 0$. The weighted matrix of an undirected graph is a symmetrical matrix with the diagonal elements being 0, while that of a directed graph is asymmetrical.

2.2. Rigid graph

As shown in Ref.[20], let $f_i(t)$ $(t \geq 0)$ denote the trajectory of the ith vertex of a formation framework. A trajectory represents the motion of a multi-agent network that is initially in the desired target formation, and an edge-consistent trajectory is defined as one such that $\|f_i(t) - f_j(t)\|$ is constant for all $(i, j) \in E$. A rigid trajectory is defined as one such that the distances between every pair of states $f_i(t)$ and $f_j(t)$ remain constant at any time t. A framework is rigid if and only if every edge-consistent trajectory of the framework is a rigid trajectory. The graph corresponding to the framework is a rigid graph. If one graph is not rigid, we say it is flexible. The rigid graphs and flexible graphs are shown in Fig.1.

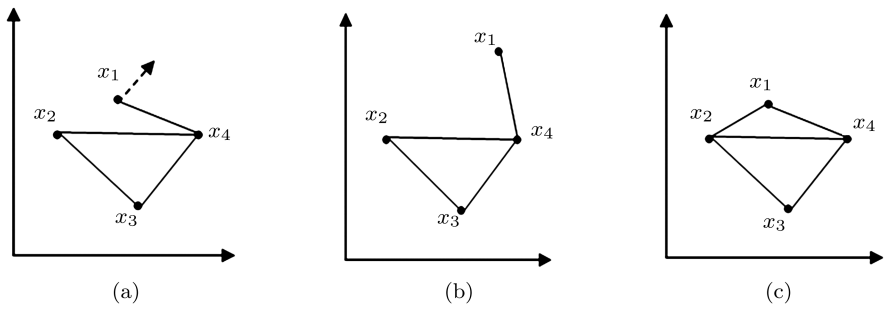


Fig.1. Flexible graph and rigid graph: (a) and (b) are flexible, while (c) is rigid.

From the graph in Figs.1(a) and 1(b), it is easy to find the theorem as follows:

Theorem 1 Every vertex in a rigid graph has at least 2 adjacent edges.

A rigid graph is not unique. Three different rigid graphs with 4 vertices are shown in Fig.2.

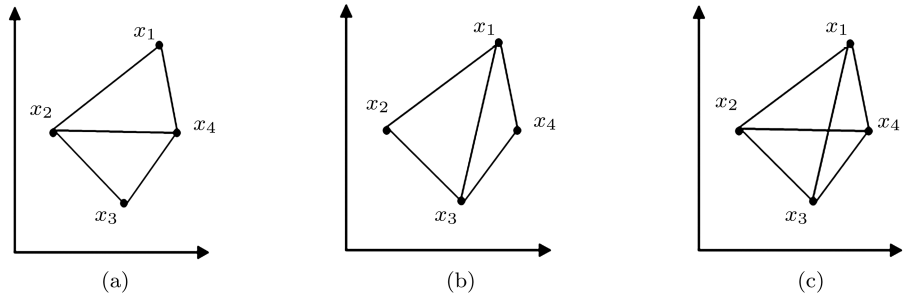


Fig.2. Three different rigid graphs with 4 vertices. (a), (b) have 5 edges, while (c) has 6 edges.

A minimally rigid graph is a rigid graph such that no edge can be removed without losing rigidity. A minimally rigid graph is not unique (i.e., the graphs (a) and (b) in Fig.2 are both minimally rigid graphs). A minimally rigid graph has some characteristics which are given as follows:

Lemma 1[14] A graph with $n \geq 2$ vertices in R^2 is minimally rigid if and only if

1) it has $2n-3$ edges and

2) each induced subgraph of $n' \leq n$ vertices has no more than $2n'-3$ edges.

2.3. Rigidity matrix

We assume that $f_i(t)$ is a differentiable function for each vertex i in a framework. If $\forall(i,j) \in E$ satisfies $\|f_i(t) - f_j(t)\|$ =constant and $(f_i(t) - f_j(t))^{\mathrm{T}}(\dot{f}_i(t) - \dot{f}_j(t)) = 0$ at $t = 0$, we say that $\dot{f} = (\dot{f}_1, \ldots, \dot{f}_n)$ is an infinitesimal flex of the framework. We say that an infinitesimal flex is trivial if it results from a rigid motion of the framework. A framework is said to be infinitesimally rigid if it has only trivial infinitesimal flexes. The notion of infinitesimal rigidity of a framework is stronger than that of rigidity.

Lemma 2[18] Infinitesimal rigidity implies rigidity.

Note, however, that rigidity does not imply infinitesimal rigidity. A more detailed description will be given after the notion of a rigidity matrix is introduced.

Given a framework with n vertices such that the position of each vertex $x_i \in R^r$ and $|E|$ edges, we order the coordinates of the vertices of a framework as

$$\{x_1^1, \ldots, x_1^r, x_2^1, \ldots, x_2^r, \ldots, x_n^1, \ldots, x_n^r\}$$

Then, we build a matrix $M \in R^{|E| \times nr}$ whose rows and columns correspond to the edges and coordinates of the vertices respectively. The elements in the row corresponding to the edge (i,j) and columns $(i-1)r+1, \ldots, ir$, $(j-1)r+1, \ldots, jr$, are $x_i^1 - x_j^1, \ldots, x_i^r - x_j^r, x_j^1 - x_i^1, \ldots, x_j^r - x_i^r$, and zero elsewhere. This matrix M is called the rigidity matrix. Tay and Whiteley[16] proved the following lemma:

Lemma 3[16] Let M be the rigidity matrix of a generic framework of n vertices in R^r. A framework with $n \geq 2$ vertices in R^2 is infinitesimally rigid if and only if

$$\mathrm{rank}(M) = 2n - 3. \tag{1}$$

Then, we use the rigidity matrix to tell the difference between rigidity and infinitesimal rigidity. From Lemma 3, the rigid graph with $\mathrm{rank}(M) = 2n-3$ is an infinitesimally rigid graph, and the rigid graph with $\mathrm{rank}(M) \neq 2n-3$ is not an infinitesimally rigid graph. An example is shown in Fig.3.

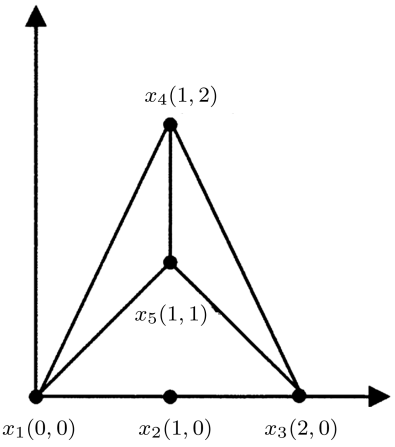


Fig.3. The framework is rigid but not infinitesimally rigid, for the rank of its rigidity matrix is $6 \neq 2 \times 5 - 3 = 7$.

In this article, we present the concept of a min-weighted rigid graph as follows:

Definition 1 If every edge of a framework $G = (V, E)$ is weighted by its length, a min-weighted rigid graph is the minimally rigid graph that has the minimally weighted sum in all infinitesimally rigid graphs.

Definition 1 shows that a min-weighted rigid graph is not only a minimally rigid graph, but also an infinitesimally rigid graph.

2.4. Persistent graph

A persistent graph is a directed graph. A persistent graph can be obtained from a relative rigid graph by adding directions for all edges. An undirected graph corresponding to the persistent graph must be a rigid graph, but we cannot generate a persistent graph by adding arbitrary directions to the edges of a relative rigid graph (see Fig.4). We define a directed graph to be a persistent graph if and only if any two vertices in the directed graph can maintain their relative distance during any motion.

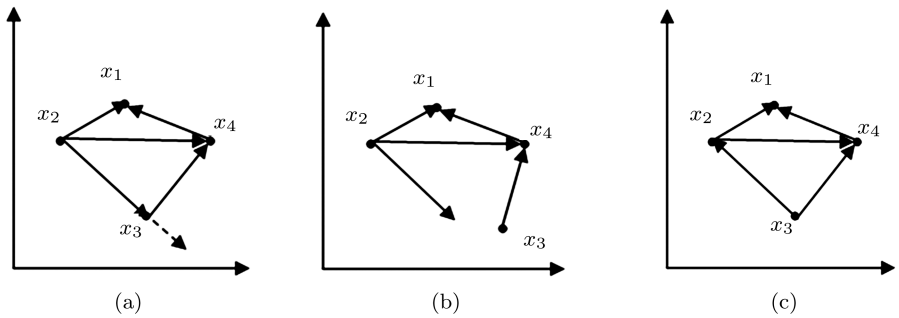


Fig.4. Persistent graph: If we add directions to the edges shown in graph (a) and let it move along the direction of the dashed arrow, the graph will change into the configuration shown in graph (b). So, graph (a) is not a persistent graph. The persistent graph (c) shows that moving any one vertex will not change the configuration.

The definition of a minimally persistent graph is as follows:

Definition 2 A graph is a minimally persistent graph if and only if it is a persistent graph and the relative undirected graph is a minimally rigid graph.

The graph shown in Fig.4(c) is a persistent graph. Persistent graphs and minimally persistent graphs are also not unique. For example, if we change the direction of the edge (x_4, x_1), the obtained graph is still a minimally persistent graph.

3. Generation of min-weighted rigidity

In this Section, we will give some theorems and the methods to generate min-weighted rigid graphs.

3.1. Generation of a minimally rigid graph

First, we show Laman's Theorem.

Lemma 4 (Laman's Theorem) $G = (V, E)$ is generically rigid in the plane if and only if there is a subset F of E satisfying:

1) $|F| = 2n - 3$;

2) For all $F' \subseteq F$, $F' \neq \phi$, $|F'| \leq 2k - 3$, where k is the number of vertices which are endpoints of edges in F'.

Smith *et al*[20] stated that "a graph is generically rigid if it has an infinitesimally rigid realization", which means that the infinitesimally rigid graph is a generically rigid graph. Note that, from Lemma 1 and Lemma 4, we will find that the notions of rigidity and generic rigidity coincide in a plane. But they do not coincide in

three or higher dimensions. This is not important in this paper and the introduction is omitted. We can obtain a theorem as follows:

Theorem 2 An infinitesimally rigid framework $G = (V, E)$ with n ($n \geq 2$) vertices and $2n - 3$ edges is minimally rigid.

Proof A framework $G = (V, E)$ is infinitesimally rigid, and therefore it is generically rigid. $|E| = 2n - 3$ implies that the subset F of E satisfying $|F| = 2n-3$ is unique, so $F = E$. From Lemma 4, for all $F' \subseteq F = E$, $F' \neq \phi$, we know that $|F'| \leq 2k - 3$, where k is the number of vertices which are endpoints of edges in F'. So, we can deduce from Lemma 1 that the infinitesimally rigid framework G is minimally rigid.

From Lemma 3 and Theorem 2, we know that if we can find a framework $G = (V, E)$ with n vertices and $2n-3$ edges, and its rigidity matrix has rank $2n-3$, then the framework is a minimally rigid graph. We consider the rigidity matrix $M_c \in R^{|E_c|\times 2n}$ of a complete plane graph $G_c = (V_c, E_c)$ with $|V_c| = n$ and $|E_c| = n(n-1)/2$. If we find a matrix $M \in R^{(2n-3)\times 2n}$ with its rows being some different rows of M_c and $\text{rank}(M) = 2n - 3$, the subgraph consisting of the n vertices and the edges corresponding to the rigidity matrix M is minimally rigid. An example is shown as follows:

Example 1 A framework in Fig.5 has 4 vertices.

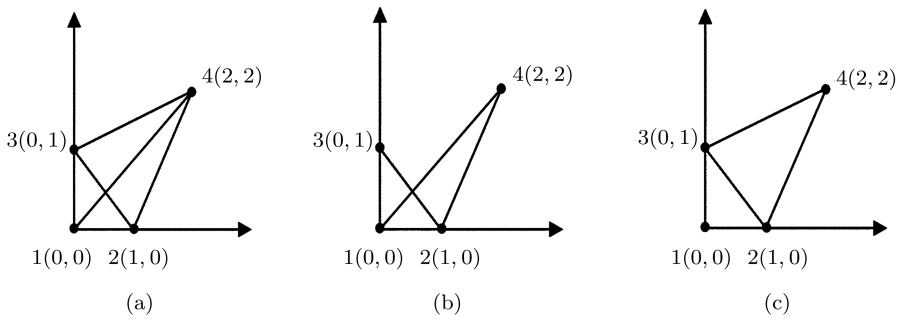


Fig.5. Graph (a) is a complete graph; graph (b) is the minimally rigid graph we find through the example; graph (c) is another minimally rigid graph which is the min-weighted rigid graph among all the minimally rigid graphs.

The rigidity matrix M_c of the complete graph (see Fig.5(a)) is

$$M_c = \begin{array}{c} \\ (1,2) \\ (1,3) \\ (1,4) \\ (2,3) \\ (2,4) \\ (3,4) \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & x_4 & y_4 \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2 & -2 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -2 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & -2 & -1 & 2 & 1 \end{bmatrix} \end{array}$$

If the number of edges of the minimally rigid graph is $2 \times 4 - 3 = 5$, considering 5 rows of M_c, we find that

$$M_c = \begin{array}{c} \\ (1,2) \\ (1,3) \\ (1,4) \\ (2,3) \\ (2,4) \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & x_4 & y_4 \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2 & -2 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -2 & 0 & 0 & 1 & 2 \end{bmatrix} \end{array}$$

and $\text{rank}(M) = 5$.

Then, the graph (as shown in Fig.5(b)) consisting of the 4 vertices and the edges (1,2), (1,3), (1,4), (2,3), (2,4) is a minimally rigid graph. In fact, using any 5 rows of M_c we can make $\text{rank}(M) = 5$, which means that using any 5 edges can form a minimally rigid graph.

It is well known that a complete graph has several minimally rigid graphs. So, from Lemma 1 and Lemma 4, it is easy to know that the complete graph is generically rigid. As stated in Ref.[20], any configuration $V = \{x_1, \ldots, x_n\}$ has the generically rigid graph $G = (V, E)$ such that G is infinitesimally rigid, which means that the complete graph with this configuration V is infinitesimally rigid. So, the rigidity matrix rank of any complete graph is $2n-3$. This shows that a min-weighted rigid graph exists (see Fig.5(c)). The methods to find a min-weighted rigid graph are shown in the following.

3.2. The methods of generating min-weighted rigid graphs

To find the min-weighted rigid graph, we will use the rigidity matrix of the complete graph. An algorithm for finding the min-weighted rigid graph will be presented in this Section, and some simulations will be given in Section 6.

The main process is described as follows:

1) Give a framework, and build a complete graph;

2) Order all the edges by their length from short to long;

3) Build the rigidity matrix of the complete graph according to the sequence of edges;

4) Calculate the rank of the matrix by adding the second row of the rigidity matrix to make a new matrix with the first row. If the matrix is of full rank, we add the second row; otherwise, delete this row from the new matrix; then, add the next row of the rigidity matrix, ..., until we find $2n-3$ linear independent row vectors.

5) Use the edges corresponding to the $2n-3$ row vectors to compose a graph. The obtained graph is the min-weighted rigid graph.

The program of generating min-weighted rigid graphs is shown in Table 1.

Table 1. Algorithm 1.

Algorithm 1: generation of min-weighted rigid graphs
Input: A framework G with n vertices which are numbered by {1,2,...,n};
Calculate all the edges of the complete graph corresponding to the framework E_c;
Sequence all the edges by increasing;
Build the rigidity matrix M_c of the complete graph according to the sequence;
Initialized M as the first row of M_c,
for 1:$n(n-1)/2$
if rank(M) <= $2n-3$
Add the next row of M_c to M to form a new matrix M';
if M' is full rank
$M = M'$;
Record the edge corresponding to the row;
end
end
end
Draw the min-weighted rigid graph;

Remark If the rigidity matrix rank of any complete graph is $2n-3$, algorithm 1 will find the graphs with $2n-3$ edges and whose rigidity matrix rank is $2n-3$. So, the graphs obtained by algorithm 1 are both infinitesimally and minimally rigid.

Theorem 3 Given a framework G with n vertices, algorithm 1 will form a min-weighted rigid graph.

Proof Let $p_1, \ldots, p_{2n-3}$ denote the $2n-3$ row vectors of matrix M obtained by algorithm 1 according to the sequence. Let $q_1, \ldots, q_m$ denote the row vectors whose row number is smaller than the row number of p_{2n-3}, but they do not belong to M. To prove the theorem, it is only necessary to prove that any change to the vectors $q_1, \ldots, q_m$ (i.e., add some vectors of $q_1, \ldots, q_m$, and delete some vectors of $p_1, \ldots, p_{2n-3}$ to form a matrix M with $\text{rank}(M) = 2n-3$) will increase the weighted sum of the framework. An easy way to prove it is to find the fact that adding any vector q_k of $q_1, \ldots, q_m$ must delete a vector p_i of $p_1, \ldots, p_{2n-3}$ sequenced before q_k. Here, we prove it by induction.

1) Add q_k only. If we delete a vector of $p_1, \ldots, p_{2n-3}$ sequenced behind q_k, q_k and all the vectors of $p_1, \ldots, p_{2n-3}$ sequenced before q_k are linear dependent (from algorithm 1, if they are linear independent, q_k would belong to M, which is not the fact). So, the rank of the new matrix will be $2n-4$ which does not fit the condition of a rigid graph. So, to maintain the rank of the new matrix, we can only delete the vector p_i of $p_1, \ldots, p_{2n-3}$ sequenced before q_k. But, the added edge corresponding to q_k is longer than the deleted edge corresponding to p_i, which means that the weighted sum will increase.

2) Add s vectors $q_{s1}, \ldots, q_{ss}$ of $q_1, \ldots, q_m$. We assume that the addition of each vector q_k of $q_1, \ldots, q_m$ must require deletion of a vector p_i of $p_1, \ldots, p_{2n-3}$ sequenced before q_k.

3) Add $s+1$ vectors. From step 2), we know that the addition of each vector $q_k \in q_{s1}, \ldots, q_{ss}$ must require deletion of a vector p_i of $p_1, \ldots, p_{2n-3}$ sequenced before q_k. Now, we consider the $s+1$ vector $q_{s(s+1)}$. If we delete the vector p_i sequenced behind $q_{s(s+1)}$, let $\{p_{s1}, \ldots, p_{ss}\}$ denote the s deleted vectors sequenced before $q_{s(s+1)}$ and $\{p_1, \ldots, p_{s+1}\}$ denote the vectors of M sequenced before $q_{s(s+1)}$. It is easy to know that each vector of $\{p_1, \ldots, p_{s+1}, q_{s1}, \ldots, q_{ss}, q_{s(s+1)} \backslash p_{s1}, \ldots, p_{ss}\}$ can be linearly expressed as $\{p_1, \ldots, p_{s+1}\}$. For the number of vectors of $\{p_1, \ldots, p_{s+1}, q_{s1}, \ldots, q_{ss}, q_{s(s+1)} \backslash p_{s1}, \ldots, p_{ss}\}$ is one more than that of $\{p_1, \ldots, p_{s+1}\}$, $\{p_1, \ldots,$

$p_{s+1}, q_{s1}, \ldots q_{ss}, \quad q_{s(s+1)} \backslash p_{s1}, \ldots, p_{ss}\}$ are linear-dependent. So, the deleted vector must be before $q_{s(s+1)}$, and the weighted sum will be increased.

In summary, the theorem is correct.

4. Generation of min-weighted persistent graph

In this Section, we will present a method based on the rigidity matrix M of the min-weighted rigid graph to generate a min-weighted persistent graph. The definition of a min-weighted persistent graph is shown as follows:

Definition 3 A graph is a min-weighted persistent graph if and only if it is min-weighted rigid and all vertices are edge-consistent.

Hendrickx *et al*[21] stated that a minimally persistent graph can be obtained from the corresponding minimally rigid graph, and gave the lemma as follows:

Lemma 5[21] For any minimally rigid graph, it is possible to assign directions to the edges such that the obtained directed graph is minimally persistent.

In this article, we can get a corollary from Lemma 5.

Corollary 1 For any min-weighted rigid graph, it is possible to assign directions to the edges such that the obtained directed graph is min-weighted persistent.

Proof A min-weighted rigid graph implies a minimally rigid graph. From Lemma 5 and Definition 3, it is easy to know that Corollary 1 is right.

From Lemma 5, we know that any rigid graph can be changed into some persistent graphs by adding different directions to the edges, but any operation needs to respect the fact that no vertex has an out-degree larger than 2. Hendrickx *et al*[15] gave the necessary and sufficient conditions for minimum persistence as follows:

Lemma 6[15] A graph is minimally persistent if and only if it is minimally rigid and no vertex has an out-degree larger than 2.

The following corollary is obvious.

Corollary 2 A graph is min-weighted persistent if and only if it is min-weighted rigid and no vertex has an out-degree larger than 2.

This is easy to deduce from Lemma 6, so the proof is omitted.

A persistent graph is not unique, and this is also true for the min-weighted persistent graph. We will use a special algorithm to find an apt min-weighted persistent graph.

The main methods are to choose a vertex as a leader in the min-weighted rigid graph, then choose a triangle which includes the leader. The configurations of the triangles with leader have two forms (see Fig.6): a) only one leader; b) three vertices are coleaders.

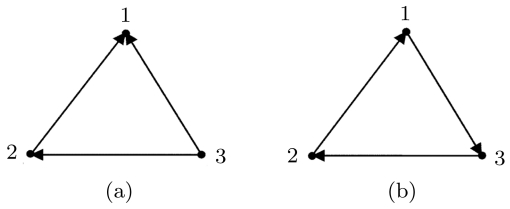


Fig.6. (a) There is only one leader; (b) There are three coleaders.

Then, each remaining vertex is assigned by two connected edges. Each edge is only assigned once, and the vertex is directed as the start point of this edge.

The method of generating min-weighted persistent graphs is shown in Table 2.

Table 2. Algorithm 2.

Algorithm 2: Generation of min-weighted persistent graphs

Input: the matrix M generated in algorithm 1 and the configuration of the triangle with leader,

Let vertex x_i be the leader,

for all the non-zero elements in the column corresponding to vertex x_i

if find two non-zero elements corresponding to (i, j) and (i, k), and row $(j, k) \in M$

break,

end

Add directions to the edges of the triangle,

set the rows corresponding to (i, j),(i, k) and (j, k) to be 0;

while $M \neq 0$

for all the rest vertices

each vertex is assigned 2 edges;

Add directions to these edges out of vertices;

The rows corresponding to these edges are set to be 0;

end

end

end

In algorithm 2, we set the rows that have been directed to be 0, ensuring that all edges are only assigned once. When $M = 0$, all the edges have been directed.

Theorem 4 Algorithm 2 generates a min-weighted persistent graph.

Proof These operations will make all the vertices with out-degree less than 2 in a triangle, and we all know that the triangle is persistent. The remaining vertices have 2 out-degrees, so they are all persistent. So the min-weighted rigid graph is persistent.

Some simulations are given in Section 6.

5. Application in formation control

A formation is a framework $G = (V, E^{\Delta})$ with the range Δ constraint. It means that if the distance of an edge (i, j) has $d_{ij} \leq \Delta$, then $(i, j) \in E^{\Delta}$, otherwise, $(i, j) \notin E^{\Delta}$. Note that not all the formations have the min-weighted persistent framework. This will be shown in the simulation results.

Theorem 5 A framework $G = (V, E^{\Delta})$ of a formation can be min-weighted persistent if and only if the rank of its rigidity matrix is $2n - 3$.

Proof (Necessity) A framework is min-weighted persistently feasible, then it is rigid. From Lemma 3, the rank of its rigidity matrix is $2n - 3$.

(Sufficiency) The rank of the rigidity matrix is $2n - 3$, so the framework is rigid. Obviously, the min-weighted rigid graph exists, so the min-weighted persistent graph exists. The theorem is proved.

Corollary 3 A framework $G = (V, E^{\Delta})$ of a formation does not exist a min-weighted persistent graph if and only if the rank of its rigidity matrix is less than $2n - 3$.

This can be deduced from Theorem 5; the proof is omitted here. Next, the algorithm of generating a min-weighted persistent formation is shown in Table 3.

Table 3. Algorithm 3.

Algorithm 3: Form a min-weighted persistent formation

Input: A framework G with n vertices that are numbered by {1,2,..., n};
Calculate all the existing edges of the formation E^{Δ};
Sequence all the edges by increasing;
Build the rigidity matrix M^{Δ} of the complete graph according to the sequence;
Initialized M as the first row of M^{Δ}
for all the edges
  if rank(M) <= $2n - 3$
    Add the next row of M^{Δ} to M to form a new matrix M';
    if M' is full rank
      $M = M'$;
      Record the edge corresponding to the row;
    end
  end
end
if rank(M) < $2n - 3$
  return "No min-weighted persistent graph exists";
end
else if rank(M) = $2n - 3$
  Draw the min-weighted rigid graph;
  Carry out **algorithm 2**;
end

6. Simulation results

In this Section, some simulations are carried out in MATLAB to demonstrate the efficiency of our research.

6.1. Generation of min-weighted rigid graphs

Here, we consider a framework with 30 vertices. These vertices are chosen randomly from a Gaussian distribution with variance of 400. Then, we obtain the graph in Fig.7 by algorithm 1.

The black points in Fig.7(a) denote the vertices. In Fig.7(b), if we count the edges, the number will be $2 \times 30 - 3 = 57$. If any one of the edges is deleted, the rigidity will be destroyed.

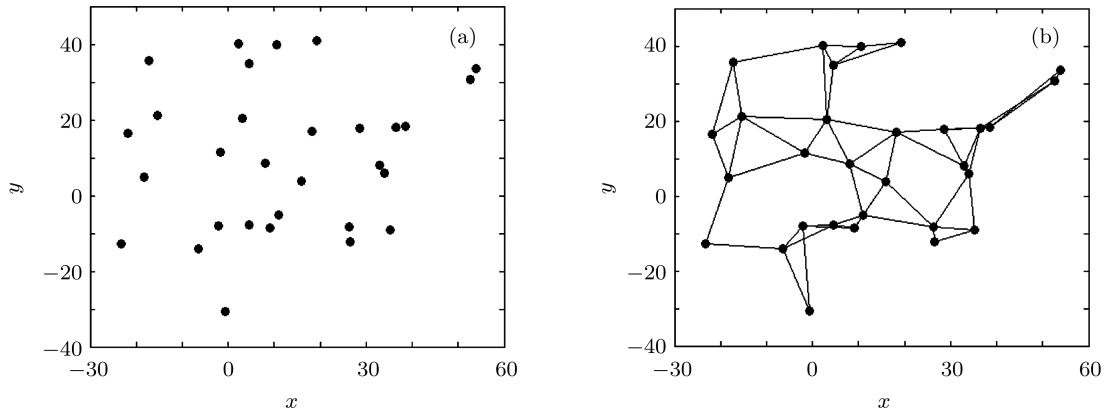


Fig.7. (a) Graph is a framework with 30 vertices which are chosen randomly from a Gaussian distribution. Graph (b) is the min-weighted rigid graph generated by algorithm 1.

6.2. Generation of min-weighted persistent graphs

For clarity, we choose a framework with 15 vertices. These vertices are also chosen randomly from a Gaussian distribution with variance of 400. The framework with no edges is shown in Fig.8(a), and the min-weighted persistent graph generated by algorithm 2 is shown in Fig.8(b).

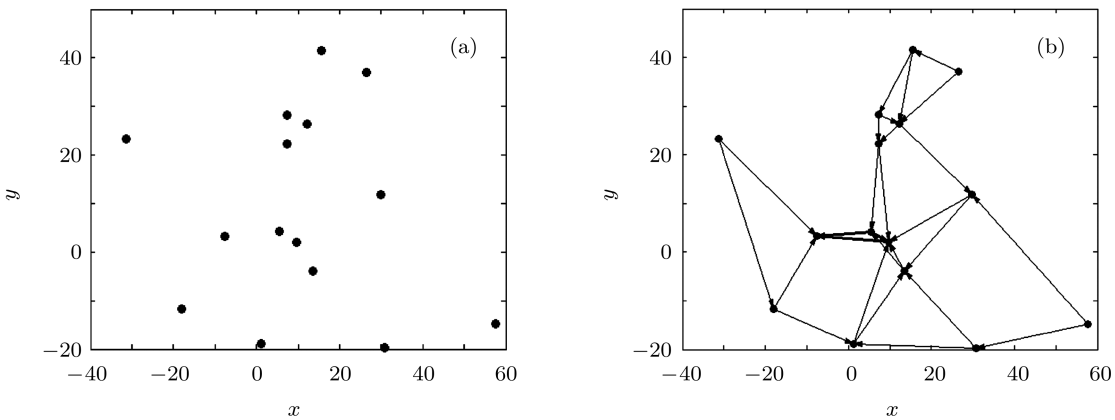


Fig.8. Graph (a) is a framework with 15 vertices which are also chosen randomly from a Gaussian distribution. Graph (b) is the min-weighted persistent graph generated by algorithm 2.

Here, we choose the configuration with only one leader. The vertex directed by 2 black-broad arrows is the leader, and the black-broad triangle is the configuration which we find firstly in algorithm 2. Apart from three vertices in the black-broad triangle, all other vertices have 2 out-degrees, which means that the min-weighted rigid graph is a min-weighted persistent graph.

6.3. Generation of min-weighted persistent formation

Here, we use another 15 vertices which are chosen randomly too. Figure 9(a) is the framework of the formation. First, we constrain the range $\Delta = 30$; the algorithm 3 returns a sentence in the command window: "No min-weighted persistent graph exists!"

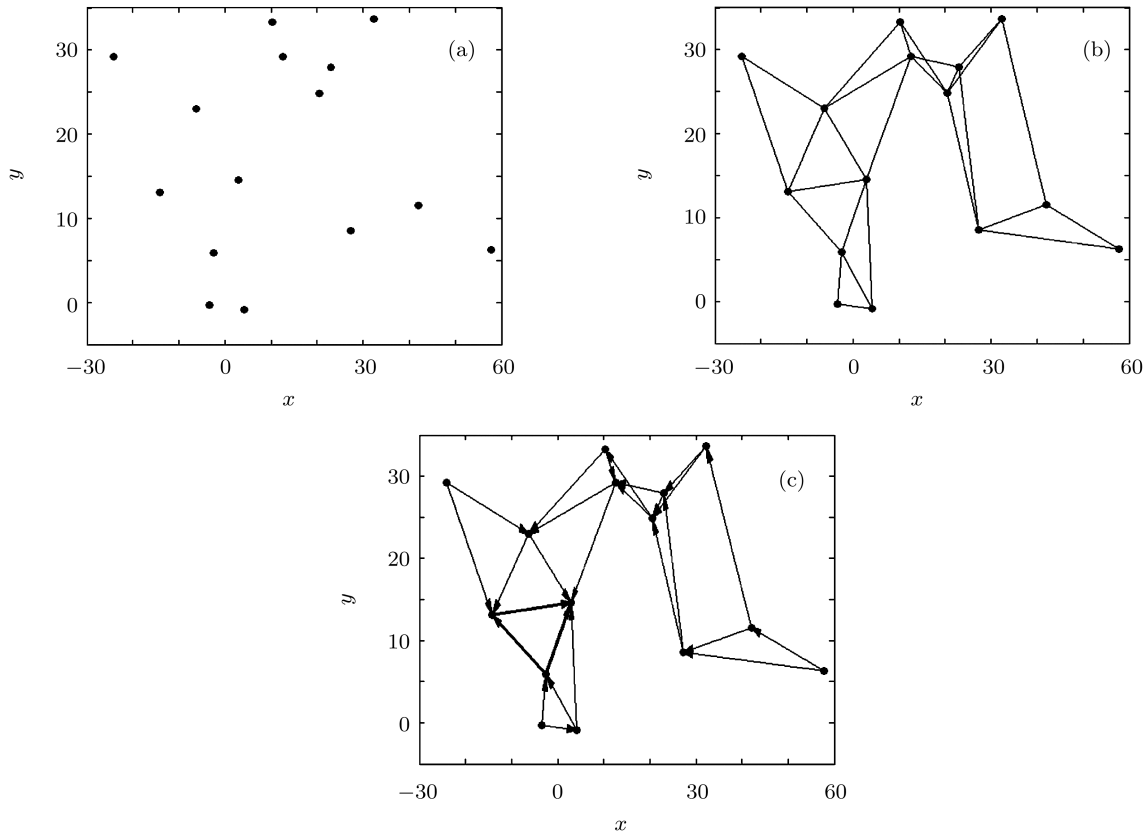


Fig.9. Graph (a) is the framework of a formation. Graph (b) is the min-weighted rigid graph of the formation when the range is 40. Graph (c) is the min-weighted persistent graph.

Then, the range is changed to 40. This time, algorithm 3 generates the min-weighted rigid graph (see Fig.9(b)) and the min-weighted persistent graph (see Fig.9(c)). The black-broad triangle is the first triangle found as described in algorithm 2.

From Fig.9(b), we will find that the line in the lower right of the min-weighted rigid graph is longer than 30 and shorter than 40, so, the min-weighted rigid graph does not exist when $\Delta = 30$ but exists when $\Delta = 40$.

7. Conclusions

In this paper, we use the rigidity matrix to generate the min-weighted rigid graph and the min-weighted persistent graph. An algorithm for generating the min-weighted rigid graph is given, and we prove that the algorithm is efficient by giving Theorem 3. Algorithm 2 shows the methods based rigidity matrix to generate the min-weighted persistent graph. Theorem 4 is given to prove that the algorithm is right. Starting from algorithms 1 and 2, we consider the formations with range constraints, and present algorithm 3 to generate the min-weighted persistent formations. Finally, three parial simulations are given to show that the three algorithms are efficient.

Our work mainly studies the formations in a plane. The algorithms in this paper are suitable for large scale formations in a plane. The future research may concern the formations in a three-dimensional space.

Acknowledgment

We thank Brian Smith, Julien M Hendrickx, *et al*, for their works have truly inspired this work. We also thank the anonymous reviewers and the editors for their valuable comments.

References

[1] Zhao J, Tao L, Yu H, Luo J H, Cao Z W and Li Y X 2007 *Chin. Phys.* **16** 3571
[2] Tan F X, Guan X P and Liu D L 2008 *Chin. Phys.* B **17** 3531
[3] Burns R, McLaughlin C A, Leitner J, Martin M, Res. Lab. and Kirtland Air Force Base, NM. 2000 *In Proc. IEEE Aerospace Conf.* **7** 19
[4] Schaub H, Vadali S R, Junkins J L and Alfriend K T 2000 *J. Astronaut. Sci.* **48** 69
[5] Buzogany L E, Pachter M and Azzo D J J 1993 *In Proc. AIAA Conf. on Guidance, Navigation, and Control* **ADA259020** 1349
[6] Wolfe J D, Chichka D F and Speyer J L 1996 In *Proc. AIAA Conf. Guidance, Navigation, and Control* **AIAA Paper** 96
[7] Curtin T B, Bellingham J, Catipovic G J and Webb D 1993 *Oceanography* **6** 86
[8] Smith T R, Hassmann H and Leonard N E 2001 In *Proc. 40th IEEE Conf. Decision and Control* **5** 4598
[9] Bender J G 1991 *IEEE Trans. Veh. Technol.* **40** 82
[10] Swaroop D and Hedrick J K 1999 *ASME J. Dyna. Syst., Measurement and Control* **121** 462
[11] Yamaguchi H 1999 *Int. J. Robot. Res.* **18** 931
[12] Yamaguchi H, Arai T and Beni G 2001 *Robot. Automat. Syst.* **36** 125
[13] Olfati-Saber R 2006 *IEEE Trans. on Automat. Control* **51** 401
[14] Laman G 1970 *J. Engineering Mathematics* **4** 331
[15] Hendrickx J M, Anderson B D O, Delvenne J C and Blondel V D 2000 *Int. J. Robust and Nonlinear Control* **17** 960
[16] Tay T and Whiteley W 1985 *Structural Topology* **11** 21
[17] Roth B 1981 *The American Mathematical Monthly* **88** 6
[18] Gluck H 1975 In *Geometric Topology, Lecture Notes in Math.* (Berlin: Springer) 225
[19] Asimow L and Roth 1978 *Trans. Am. Math. Soc.* **245** 279
[20] Smith B, Egerstedt M and Howard A 2007 *ACM Int. Conf. Proc. Series* **318** 1
[21] Hendrickx J M, Fidan B, Yu C, Anderson B D O and Blondel V D 2006 *In Proc. of the Math. Theory of Networks and Systems (MTNS) Conf.* **17** 859
[22] Hendrickx J M, Fidan B, Yu C, Anderson B D O and Blondel V D 2008 *IEEE Trans. on Automatic Control* **4** 968
[23] Godsil C and Royle G 2001 *Algebraic Graph Theory, Gruduate Texts in Mathematics* (New York: Springer-Verlag) p1