

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

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

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

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

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Graph-based Multi Robot Motion Planning: Feasibility and Structural Properties

Ellips Masehian and Azadeh H. Nejad
Tarbiat Modares University
Iran

1. Introduction

Future generation autonomous agents are expected to operate in remote and dangerous places like outer space, undersea, hazardous waste sites, and are therefore anticipated to be far more self-directed than today's existing agents. The ability of an agent to plan its own motion is pivotal to its autonomy. For over more than three decades, agent motion planning in general, and robot motion planning in particular, have attracted much research in various fields and have become central topics in autonomous agents and artificial intelligence. Although today the term 'motion planning' is considered to cover a wide variety of problems, we will use it for the problem of planning collision-free motions for an autonomous robot moving among obstacles.

The necessity of planning the motions of autonomous agents originally arose in early 1970's, when the first industrial robots were to perform automatic tasks of manipulation and navigation. Soon it was realized that the complexity of the robot motion planning problem is PSPACE-hard and NP-complete since the size of the solution space grows exponentially and gets extremely complicated, especially for high degrees of freedom (Canny, 1988).

Many techniques have been developed for solving the robot motion planning problem, including the 'Skeleton' or 'Roadmap' approach (Choset et al., 2005). In this approach, the continuous workspace is mapped into a one-dimensional graph with vertices including the start and goal of the robot, and edges as paths between vertices. This graph is then searched to find a collision-free start-to-goal path.

Visibility Graph is a type of roadmap which is the collection of lines in the free space connecting vertices of an object to those of another (Fig. 1(a)). There are $O(n^2)$ edges in the Visibility Graph, and it can be constructed in $O(n^2)$ time and space in 2D, where n is the number of vertices.

Voronoi Diagram is another roadmap defined as the set of points equidistant from two or more objects (Fig. 1(b)). The Voronoi Diagram partitions the space into regions, where each region contains one object. For each point in a region, the object within the region is the closest to that point than any other object. There are only $O(n)$ edges in the Voronoi Diagram, and it can be efficiently constructed in $\Omega(n \log n)$ time, where n is the number of objects (Hwang & Ahuja, 1992).

When multiple moving robots share a common workspace, the motion planning task becomes even more difficult and cannot be performed for just one robot without considering others. In this kind of problems, while pursuing their individual (local) goals,

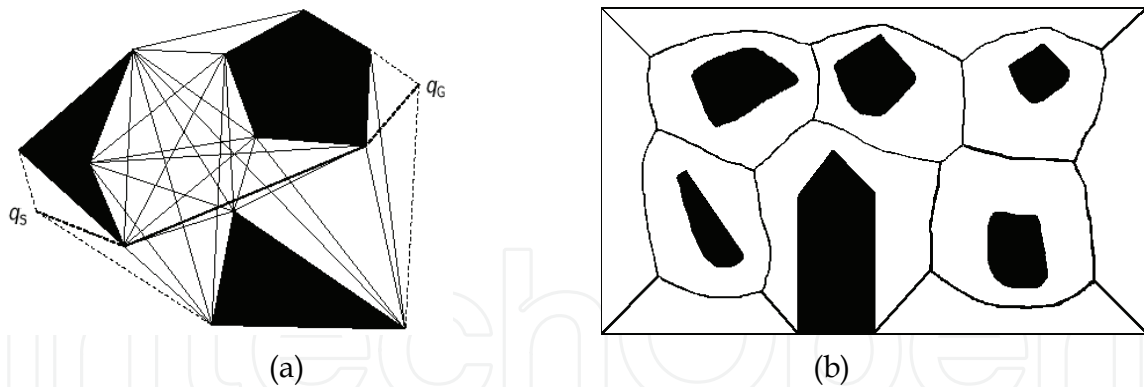


Fig. 1. (a) Visibility Graph, and (b) Voronoi Diagram

robots must coordinate their motions with each other in order to avoid collisions with obstacles and one another, thus contributing to the task of achieving a global goal, which might be minimizing the total time or distance. This problem is called Multi Robot Motion Planning (MRMP) problem. In MRMP, each robot is regarded as a dynamic obstacle for other robots, and therefore the element of time plays a major role in planning, especially because of its irreversible nature (Latombe, 1991).

Although classic roadmaps like Visibility Graph and Voronoi diagram have proved to be effective for single-robot problems, they do not provide straightforward solutions to MRMP problems, including the corridor-like environment shown in Fig. 2(a). This kind of environments is prevalent in large warehouses, plants, and transportation systems, where Automatic Guided Vehicles (AGVs) convey material and products (Fig. 2(b)).

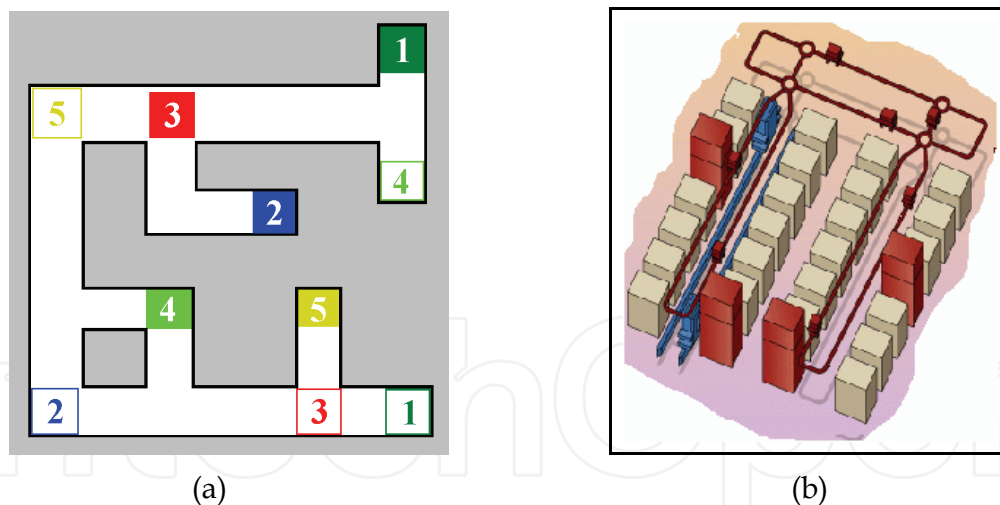


Fig. 2. (a) Each robot must move toward its final destination while allowing other robots to reach their goals. (b) Planning the motions of AGVs to different locations in a shop floor or warehouse is a real-world application of MRMP

Space is the most limiting constraint in a typical MRMP problem: often, because of lack of sufficient space around moving robots, they cannot reach their destinations without obstructing each other's way, causing deadlocks. Deadlocks are situations in which two (or more) robots intercept each other's motions and are prevented from reaching their goals. This happens generally in narrow passageways where autonomous moving robots cannot pass by each other. To resolve such a deadlock, one of the robots should leave and evacuate

the passageway (by usually backtracking), and let the opposite robot move out of the passage.

A well-known cooperative behaviour of robots is following independent start-to-goal paths while resolving deadlocks by reshuffling, circumnavigating, detouring, and speed regulating, which is also known as Velocity Tuning and is first developed in (Kant & Zucker, 1986). Another approach developed for resolving deadlocks is the Prioritized Planning, in which the robots are sorted by their moving priorities (Bennewitz et al., 2001). Higher-priority robots are planned first, whereas lower-priority robots plan their motion subsequently by considering higher-priority robots as dynamic obstacles.

By reducing the workspace into a graph with vertices including the starts and goals of all robots, the MRMP problem can turn into a sequencing problem where the robots are planned to move sequentially (or concurrently) toward their destinations, without colliding with each other. The graph structure stipulates them to remain on predefined routes (i.e. graph edges), and so avoid static obstacles existing in the workspace.

The main question in designing a predefined graph, however, is to find out whether the graph is 'reachable' (solvable) for any initial and final configurations. Solvable Graphs allow the transition of any initial configuration of agents (e.g. pebbles (beans), robots, or vehicles) to a final state via their sequential moves along the graph's edges.

Wilson in (Wilson, 1974) worked out a relation between the number of pebbles (k) and the number of vertices (n) of only bi-connected graphs as $k = n - 1$. Kornhauser in (Kornhauser et al., 1984) improved this result through generalizing the decision problem for all graphs and any number of agents. (Auletta et al., 1999) studied the above problem as pebble motion problem by following the generalization of the 15-puzzle and presented a linear algorithm for deciding the reachability of trees. Ryan in (Ryan, 2008) studied the possibility of reaching destinations of connected sub-graphs by simplifying the multi robot motion planning between the sub-graphs. He worked on predefined sub-graphs like stack, clique and hall. In (Onn & Tennenholtz, 1997) it is demonstrated that the environment can be shown through any two-connected graph which has a routing with a practical social law for motion planning.

To our knowledge, the problem of deciding whether a graph is always solvable for a specific number of robots for any initial and final configuration has never been mentioned or addressed in the literature prior to our work (Masehian & Hassan Nejad, 2009). The problem of determining the smallest solvable graph (in terms of vertices) for a certain number of robots has also been remained untackled.

Graphs can be categorized into two general classes: cyclic, and acyclic: cyclic graphs have loops and are more convenient for moving of multiple robots (or agents), while acyclic graphs (i.e. trees) provide less manoeuvrability for the agents moving on it. That's why MRMP on trees can serve as a basis for MRMP on general cyclic graphs. MRMP on trees has real-world applications in maze-like environments, indoor corridors, parking lots, railway networks, etc.

In this chapter, we specifically deal with tree-type (acyclic) graphs, and set forth the following questions: What is the maximum number of robots a tree can accommodate such that any final configuration can be reached from any initial configuration? What topology a tree must have to be solvable for a specific number of robots? and What is the 'smallest' tree solvable for a specific number of robots?

After presenting some basic definitions in section 2, in section 3 the conditions for a tree to be solvable are investigated and the maximum number of robots located on it is determined.

The concept of Minimal Solvable Trees is introduced in Section 4, and a practical shop-floor problem is presented in Section 5 to demonstrate the application of theoretical results. Some discussions and conclusions are presented in Section 6.

2. Definitions and assumptions

As mentioned earlier, reducing (or mapping) the configuration space into a graph is very helpful regarding the significant savings in required time and memory. In order to lay a proper mathematical foundation for expressing and investigating the properties of graphs, we adopt the standard terminology used in Graph Theory (Diestel, 2000). In addition, some definitions and symbols have been introduced and defined specifically for this work, all presented in Table 1. A number of these concepts are illustrated in Fig. 2.

Generally, an MRMP in a continuous space is composed of the following phases:

- a. Constructing a network (graph) with nodes (vertices) including the robots' initial and final stopping locations, together with all the locations the robots should be able to temporarily stop at and offer a service. The arcs (edges) of the graph must pass through free spaces (e.g. unoccupied rooms or corridors) such that no obstacle may be collided with during the robots' motions along the edges.
- b. Planning the robots' motions along the arcs of the network from their starting to final locations, such that a cost criterion (e.g. time, distance, or expense) is minimized.

Term/Symbol	Description
Order, $ G $	The number of vertices of the Graph $G = (V, E)$.
$\ G\ $	The number of edges on G .
Path	A non-empty graph $P = (V, E)$ of the form $V = \{v_0, v_1, \dots, v_k\}$, $E = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k\}$, where all v_i 's are distinct.
Cycle	A non-empty graph of the form $V = \{v_0, v_1, \dots, v_k\}$, $E = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k, v_kv_0\}$, where all v_i 's are distinct.
Cycle Edge	An edge on a cycle.
Tree	An acyclic graph.
Leaf, L	A vertex with a degree $d = 1$.
Cycle Vertex, C	A vertex located on a cycle.
Internal Vertex, I	A vertex with a degree $d > 1$ which is not a Cycle Vertex.
Stem, S	The longest path in the graph with its one end (or both ends, if located between two cycles) connected to a cycle vertex and including it, and not containing any Cycle Edge. None of the edges of the Stem are cycle edges. If not unique, the Stem is selected arbitrarily. The number of vertices on the Stem (i.e. its order) is shown by $ I_S $.
Configuration	An arrangement of robots on the graph vertices such that no vertex is occupied by more than one robot.

Table 1. Definitions of used terms and symbols

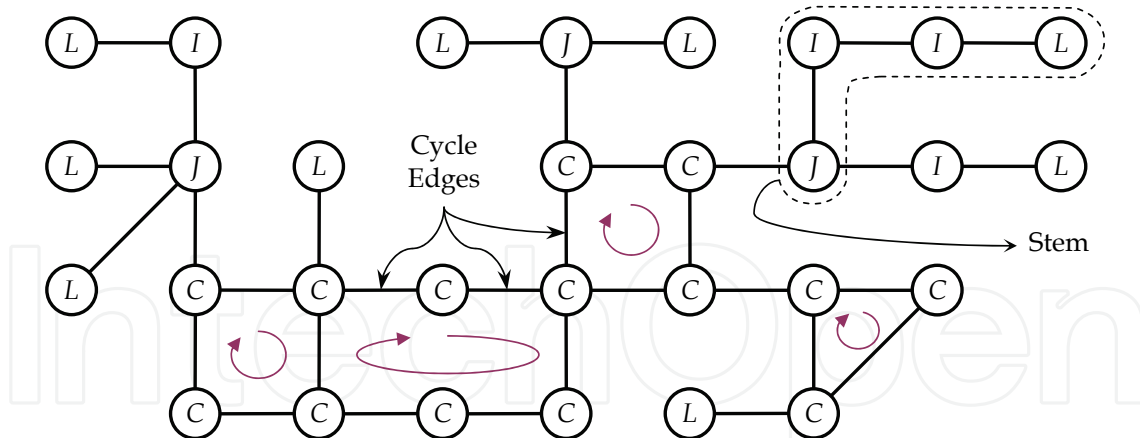


Fig. 3. Some concepts illustrated: L , I , J and C denote Leaf, Internal vertex, junction and Cycle-vertex, respectively. The symbol \curvearrowright indicates a cycle, and the dashed area designates the Stem

When designing a graph or network of routes, it is always important to consider current transportation demands, as well as future developments of the system. In the context of MRMP, this consideration requires that the system designer decides the proper topology of the network, the number of agents (as mobile robots or vehicles) required to move along the routes, and the possibilities of expanding the network for future increased transportation traffic.

Concerned about these issues, we have comprehensively investigated the concept of Solvable Graphs. To date, the notion of graph solvability has been essentially dependent to the initial and final configurations (situations) of the moving objects. For instance, the question whether a tree-like graph is solvable for a given initial and final configuration of pebbles is solvable or not is addressed in (Auletta et al., 1999). However, no work exists in the literature for all types of graphs, and never has the problem of deciding if a graph is always solvable for a specific number of robots for any initial and final configuration been mentioned or addressed.

In this chapter, we just focus on the first phase of MRMP, that is, graph construction, and propound some related problems of broad scope, such as:

- What is the maximum number of robots a graph can accommodate such that any final configuration can be reached from any initial configuration?
- What topology must a graph have to be solvable for a specific number of robots?
- What is the 'smallest' graph solvable for a specific number of robots?

Before dealing with the answers to the above questions, three new fundamental and correlated notions are presented:

Definition 1. A Solvable Graph is a graph on which any configuration of at most m robots can be reached from any initial configuration through their moves on graph edges, and is shown by SG^m .

Definition 2. A Partially Solvable Graph is a graph on which only some configurations of m robots can be reached from any initial configuration through their moves on graph edges, and is shown by PSG^m .

Definition 3. A Minimal Solvable Graph is the smallest graph on which any configuration of at most m robots can be reached from any initial configuration through their moves on graph edges, and is shown by MSG^m . In this definition, 'smallest' can be expressed and measured in terms of the number of either vertices or edges.

Graphs can be categorized into two general classes: (1) Cyclic graphs, and (2) Acyclic graphs. Cyclic graphs have loops and provide alternative ways to access a specific node, and therefore are more convenient for planning the motions of multiple robots. On the other hand, Acyclic graphs (i.e. trees) do not have any loops, and the shortest path between every pair of vertices is unique. As such, trees provide fewer options and less manoeuvrability for robots, and are much harder to solve than cyclic graphs. Actually, MRMP on trees can serve as a basis for MRMP on cyclic graphs, and the solution of an MRMP on a tree is also valid for the MRMP problem on a cyclic graph which is the superset of that tree. In view of this, and considering its NP-hardness, we found the MRMP problem on trees a worthwhile and challenging problem to be solved efficiently.

2.1 Assumptions

In the phase of graph construction, the topological (and not geometric) features of the world are important; features like the number and degrees of vertices, existence of edges between certain pairs of vertices, existence of loops and their sizes, etc. Nevertheless, the geometrical features of the world, such as the exact coordinates of the vertices, the lengths of edges, etc. become decisive in the phase of motion planning.

In this chapter some simplifying (yet not limiting) assumptions about the graph and robots are made as following:

1. An essential assumption is that the designed graph is finite, connected, planar, acyclic, and represents the free space. This means that edges intersect only at vertices.
2. The graph is undirected, and a path exists from any vertex v to u and vice versa.
3. The initial and final locations of all robots lie on the graph and are known.
4. All robots share the same graph and can (and may) move on the edges of the graph and stay on the vertices of the graph. A Move is defined as transferring a robot from a vertex to its neighbouring vertex via their connecting edge.
5. Two or more robots may not simultaneously occupy the same vertex in the graph. That is, the vertices are supposed to be spaced sufficiently far apart so that two robots can occupy any two distinct vertices without having collision.
6. Robots have sequential (i.e. one at a time) movements on edges of the graph. In other words, a robot at vertex v can move to its neighbouring Leaf or Internal vertex u only if u is unoccupied. Robots occupying other vertices in the graph do not affect this movement.

3. Solvable trees

Since junctions are vertices in the tree where different branches and Leaves meet (just as squares or crossroads which connect avenues and streets), they enable the robots to change their course of motion and shift from one branch or Leaf to another branch or Leaf.

Therefore, Leaves and junctions' branches can be used for robots' manoeuvres and interchanges, and serve as places for situating the robots permanently or temporarily during the motion planning task, aiming to make the start-to-goal paths of robots as free as possible and facilitate the robots' moves toward their goals.

On the other hand, the exchangeability of robots at and around a junction also depends on the number of vertices not occupied by robots at the initial configuration. We call these empty vertices 'Holes', and their number, H , is calculated by $H = |G| - m$, where $|G|$ is the order of the graph.

In this section, for obtaining the solvability conditions of trees, first the simplest trees called ‘Stars’ are introduced and their solvability is investigated. The results are then generalized to more complicated trees. It is noted that we are trying to find the *maximum* number of robots for which a tree is solvable. Obviously, any tree solvable for m robots is also solvable for $k < m$ robots since there will be more empty vertices and so deadlocks can be resolved more easily.

The basic requirement for the solvability of a graph is expressed as follows:

Lemma 1. *A graph is SG^m iff any two robots located on it can exchange their positions.*

Proof. Suppose that m robots have to move from their initial positions to specified final positions. We represent the exchange of two robots r_i and r_j by $X(r_i, r_j)$ and denote the act of moving a robot r_i to an empty vertex by $M(r_i, h)$, and assume that robots’ exchanges are possible while the initial and final positions of all other robots remain unchanged. It is noted that while the premise of the lemma deals with exchanging of two robots, moving a robot to an empty vertex is a direct ramification of it since it is equivalent to exchanging a real robot with a dummy one.

The planning of start-to-goal motions of all robots can be decomposed into a sequence of 2-robot exchanges, as follows:

Stage 1:	$M(r_1, h)$	or	$X(r_1, r_i),$	$\forall i \in \{2, 3, \dots, m\},$
Stage 2:	$M(r_2, h)$	or	$X(r_2, r_i),$	$\forall i \in \{3, 4, \dots, m\},$
...				
Stage $m-1$:	$M(r_{m-1}, h)$	or	$X(r_{m-1}, r_i),$	$\forall i \in \{m\},$
Stage m :	$M(r_m, h).$			

Note that for accomplishing each Stage numerous steps (i.e. moves) might be necessary. It follows that if according to the premise of the lemma any 2-robot (either real or dummy) exchange is possible, then any m -robot exchange is also possible, which means that the graph is solvable.

Conversely, if a graph is solvable, then any configuration is reachable from any initial configuration—a special case of which could be the exchanging of just two robots. This shows that graph solvability and 2-robot exchanging are logically equivalent. □

3.1 Solvability of star trees

A $Star^k$ is a complete bipartite graph with only one vertex in one part and k vertices in the other part. Stars are trees with diameters equal to 2, and have only one junction. Fig. 4(a) illustrates a 6-Leaf Star.

An *Extended Star* (ES^k) is a graph comprised of a single junction connected to k vertices, some or all of which are connected to ‘chains’ of internal vertices, such that there are no other junctions in the graph. A typical Extended Star is shown in Fig. 4(b).

Lemma 2 investigates the solvability of Stars:

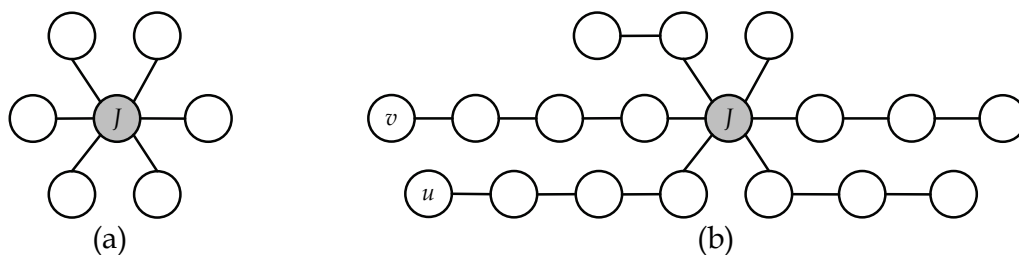


Fig. 4. (a) A 6-leaf Star, and (b) an Extended Star with $\|S\| = 4$

Lemma 2. A Star with k leaves is Solvable iff $H \geq 2$, where H is the number of Holes.

Proof. A $Star^k$ has $k + 1$ vertices (as can be seen in Fig. 4(a) for $k = 6$), and when $H = 2$, these two Holes may either be (i) on two leaves, or (ii) on a leaf and the junction. Now for proving the solvability of the Star, various possibilities of initial and final configurations of robots can be considered in the form of four Scenarios enlisted in Table 2.

	G1: The goals of all robots are on leaves	G2: The goals of all robots are on leaves except for one that is on the junction
S1: The starts of all robots are on leaves	Scenario 1	Scenario 2
S2: The starts of all robots are on leaves except for one that is on the junction	Scenario 3	Scenario 4

Table 2. Possible scenarios for robots' configurations on Stars

Scenario 1: In this scenario a Leaf and the junction are initially empty (case (ii)). If the empty Leaf is the goal of a robot, then it should be occupied by that specific robot either directly (if there is a free path for the robot), or indirectly (after freeing the path by other robots). So the Star is solvable.

Scenario 2: In this scenario the case (i) holds again. The robots should occupy their goals in the same manner as in the Scenario 1, with the consideration that occupying the junction should be the last move. The Star is therefore solvable.

Scenario 3: In this scenario, initially one robot is on the Star's junction and $m-1$ robots are on the Leaves, and so the case (i) holds true. Through a single move of the robot located on the junction to an arbitrary empty Leaf, the Scenario 1 is attained, and the subsequent moves for solving the problem can be done in the same manner.

Scenario 4: The case (i) applies for this scenario. Again by moving the robot on the junction to an empty Leaf, this scenario converts to the Scenario 2, and the problem can be solved accordingly.

If $H > 2$, then the situation for the solvability of the Star is more favourable and the Star is definitely solvable. Conversely, if $H < 2$, then the robots located on leaves can just move to the junction and cannot locate on another leaf, so the graph is not salvable. \square

Lemma 3 deals with the solvability of Extended Stars:

Lemma 3. An Extended Star is solvable iff $H \geq \|S\| + 1$, where H is the number of Holes and $\|S\|$ is the length of the graph's Stem.

Proof. According to the Lemma 1, in order for the graph to be solvable, any two robots must be able to exchange their positions. On the other hand, the worst case of exchanging occurs when a robot r_i located on the vertex v with the maximum distance to the junction, (i.e. on the leaf of the graph's Stem) has to exchange its position with robot r_j located on the furthest vertex u of the second longest chain (with a length of at most $\|S\|$) as shown in Fig. 4(b). Since there are no cycles or other junctions in the graph, the only way for robot r_i to get to another chain of the graph is to reach the junction (as proved in the Lemma 2) and then enter the destination chain. This is possible only by evacuating the path connecting the vertex v to the junction (if the path is not initially empty) plus an additional vertex

connected to the junction, by removing all obstructing robots onto the graph's empty vertices. The additional empty vertex beyond the junction serves as a 'parking', so that after locating the robot r_i on it, the destination chain can be evacuated completely by moving its robots toward the empty Stem through the junction. This means that at least $\|S\| + 1$ vertices in the graph should be empty.

Conversely, if $H < \|S\| + 1$, the robot r_i could at most reach the junction, and not farther, toward other chains. Thus the graph would be unsolvable. \square

Regarding the pivotal role of junction vertices in solvability of graphs, at this point we present a new concept, the *Influence Zone (IZ)* of a junction as the set of vertices and edges around a junction. Mathematically:

Definition 4. The Junction Influence Zone (IZ_J^i) of the junction J^i is the subgraph satisfying the following condition:

$$IZ_J^i = \{ (v, e) \in (V, E) \mid \text{Dist}(J^i, v) \leq H - 1 \} \quad (1)$$

which means that IZ_J^i contains the junction J^i plus all vertices with distances to J^i less than or equal to $H - 1$, together with all their connecting edges.

Fig. 5 illustrates Junction Influence Zones of a tree with $H = 3$ holes. In this example, the vertices in Influence Zones are: $IZ_J^1 = \{J^1, J^4, v_3, v_4, v_7, v_8, v_9\}$; $IZ_J^2 = \{J^2, v_1, v_2, v_6, v_{10}, v_{11}, v_{12}, v_{16}\}$; $IZ_J^3 = \{J^3, v_{13}, v_{14}, v_{15}, v_{17}, v_{18}\}$; $IZ_J^4 = \{J^4, J^1, v_7, v_{11}, v_{12}, v_{13}, v_{14}\}$.

By comparing the definitions of the Extended Star and Junction Influence Zone, and regarding the results of the Lemmas 2 and 3, the following corollary can be proposed about the general condition for a Junction IZ to be solvable:

Corollary 1. Any Junction Influence Zone having at least $\|S\| + 1$ Holes is solvable.

Proof. Considering the following two facts:

- i. Junction Influence Zones (IZ_J) may contain multiple junctions whereas Extended Stars (ES) do not contain any junction (other than the central junction, as seen in Fig. 4(b)); and,
- ii. Junctions in a graph provide possibilities for robots' exchange and temporary residence,

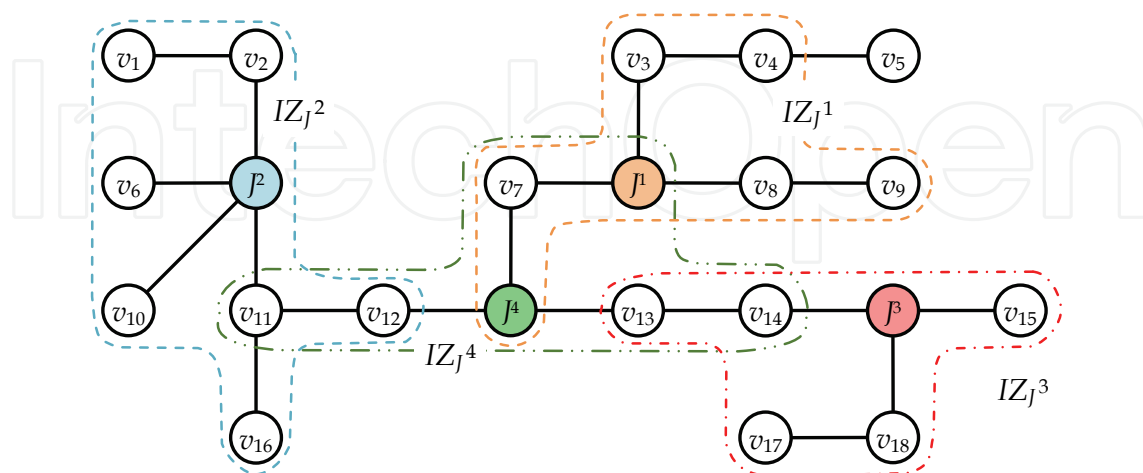


Fig. 5. Junction Influence Zones of a tree with $H = 3$ holes

it is concluded that the topology of an IZ_J with just one junction is analogous to that of an ES, and since there are less possibilities in ES for robot's exchanges (due to the existence of

just one junction), the condition for an ES to be solvable is at least as hard as the condition for an IZ_J to be solvable. In other words, if an ES with at least $H = \|S\| + 1$ Holes is solvable, then a Junction IZ_J with at least $H = \max\{Dist(J^i, v) + 1; \forall v \in V\} = \|S\| + 1$ Holes would also be solvable. \square

3.2 Solvability of general trees

A general tree usually has multiple junctions, and so can be partitioned into a number of zones formed around each junction, based on the Definition 4. Note that this decomposition is unique for any tree with a specified number of Holes.

After determining the Junction Influence Zones of a general tree, its solvability can be assessed in two phases:

1. checking the possibility of robots' exchanges within a zone, and
2. verifying the possibility of robots' exchanges between multiple zones.

The first phase (i.e. the solvability of a Zone) was discussed in the Corollary 1, the results of which are now generalized to the general trees. In this subsection, we will specify the conditions for global exchanges of robots, based on which the solvability of general trees can be proved.

As mentioned in the Lemma 1, all robots can access all vertices in a solvable graph. In fact, in the second phase we intend to verify the condition for a typical robot to expand the scope of its movements beyond the limits of the Influence Zone(s) within which it is located. Such an expansion of movements would essentially start with 'penetrating' into *Adjacent* zones. The concept of Adjacency is defined in Definition 5:

Definition 5. Any two junctions (J^p and J^q) are considered Adjacent if there is no other junction (J^r) between them. That is,

$$\langle J^p \leftrightarrow J^q \rangle \xleftrightarrow{\text{if}} \langle \nexists J^r \in Path(J^p, J^q); \quad \forall J^p, J^q, J^r \in J(G) \rangle, \quad (2)$$

in which the symbol ' \leftrightarrow ' stands for Adjacency. In other words, two junctions are Adjacent if there is no other junction on their connecting shortest path. Two Influence Zones are considered as *Adjacent* if their junctions are Adjacent (see Fig. 6).

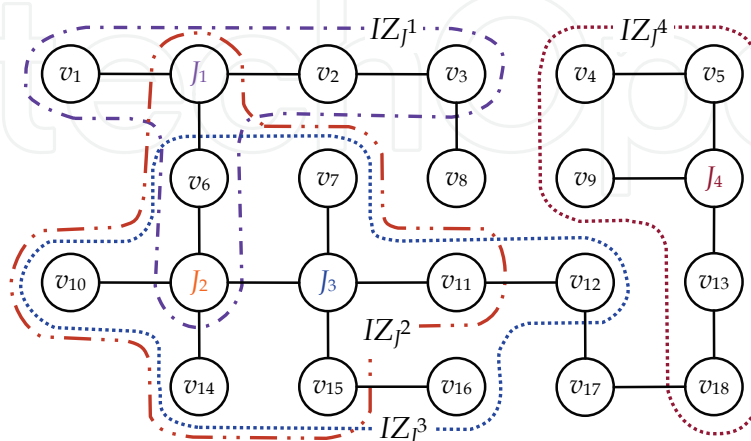


Fig. 6. Influence Zones of the junctions of a sample tree with $H = 3$. Note that v_8 and v_{17} do not belong to any IZ.

In addition to the Adjacency concept, the new notion of *Interconnectedness* is also introduced here to work out the solvability conditions of a general tree (i.e. with multiple Influence Zones), as defined in Definition 6:

Definition 6. Two Influence Zones (IZ^p and IZ^q) are Interconnected if

$$\langle IZ^p \equiv IZ^q \rangle \xleftrightarrow{\text{if}} \langle IZ^p \leftrightarrow IZ^q \rangle \wedge \langle \text{Dist}(X^p, X^q) \leq H - 2; X^p, X^q \in J(G) \rangle, \quad (3)$$

In which the symbol ' \equiv ' indicates Interconnectedness. This means that the maximum distance between the junctions of two Influence Zones must be at most $H - 2$. Note that every two Interconnected IZs are Adjacent, but the reverse is not true. For instance in Fig. 6, $IZ_1^1 \equiv IZ_2^2$, and no other IZs are Interconnected.

Now the conditions for exchangeability of any two robots located in Interconnected Influence Zones are investigated, which will lead to the solvability of general trees. Obviously, the results for Interconnected zones can be generalized to trees with multiple zones due to the transitive property of sets (in this case, IZs). Theorem 1 deals with the conditions for exchanges of robots in Interconnected Influence Zones:

Theorem 1. Any robot located on an Influence Zone can access any vertex in its Adjacent Influence Zone if the two Zones are Interconnected.

Proof. We will prove the theorem by indirect proof. Suppose that the two Adjacent Influence Zones IZ^p and IZ^q are not interconnected. We will prove that this supposition is not true.

If the two Influence Zones are not Interconnected, then $\text{Dist}(J^p, J^q) > H - 2$, being at least $(H - 1)$. Fig. 7 shows two Adjacent IZs with $H = 5$ and $\text{Dist}(J^p, J^q) = 4$. The worst case of accessibility occurs when the robot on vertex 1 wants to access the vertex 7. To make this possible, the path connecting the two junctions (inclusive) and the destination vertex 7 should be empty (or able to be emptied by removing all the robots occupying the path); that is, vertices 2, 3, 4, 5, 6, and 7 in the figure.

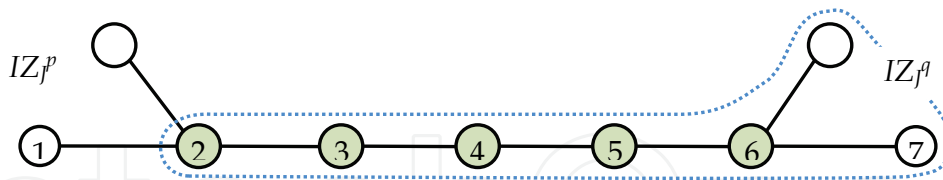


Fig. 7. Illustration for the proof of Theorem 1. Dashed areas designate Influence Zones, and dark vertices show Holes.

Therefore, the required number of Holes would be:

$$H^* = | \text{Path}(J^p, J^q) | + 1 = H + 1 \quad (4)$$

in which $| \text{Path}(J^p, J^q) | = \text{Dist}(J^p, J^q) + 1$. Since (4) is in contradiction with the assumption of existing H Holes, it is concluded that $\text{Dist}(J^p, J^q) \leq H - 2$, in which case IZ^p and IZ^q will be Interconnected according to (3). \square

By combining the results of the Theorem 1 and the Lemma 1, it is possible to work out the solvability conditions of any general compound graph with multiple influence zones, as exhibited in Theorem 2. Note that this theorem also holds true for graphs with a single influence zone.

Theorem 2. Any graph with multiple Influence Zones is solvable if and only if the following conditions are satisfied:

- i. All two Adjacent Influence Zones are Interconnected,
- ii. All vertices belong to at least one Influence Zone.

Proof of ‘if’. We will first prove the following statement: “A:= if the conditions (i) and (ii) hold, then a graph is solvable” by direct proof. Regarding the Lemma 1, the solvability of a graph is equivalent to the possibility of exchanging any two robots on it while the arrangement of all other robots remains unchanged. Therefore, alternatively, we want to prove that “A’:= if the conditions (i) and (ii) hold, then any two robots can exchange while the arrangement of other robots remain unchanged”.

If an arbitrarily selected robot is located on any vertex, then this vertex belongs to at least one IZ (regarding to (ii)), and the robot not only can move to anywhere in its IZ (regarding Corollary 1), but also to any vertex in its Interconnected IZ (according to the Theorem 1).

We define the expression $X\left(\frac{r_i}{IZ^m}, \frac{r_j}{IZ^n}\right)$ as the act of exchanging the positions of robots r_i and r_j located in the IZ^m and IZ^n Influence Zones, respectively. Now suppose that a graph has k Influence Zones with the order of IZ^1, IZ^2, \dots, IZ^k being mutually interconnected, as affirmed by the condition (i). It can be shown straightforwardly that the $X\left(\frac{r_1}{IZ^1}, \frac{r_k}{IZ^k}\right)$ can be expressed by a series of consecutive intermediate exchanges between two Adjacent IZs, as follows:

$$(IZ^1 \equiv IZ^2) \wedge (IZ^2 \equiv IZ^3) \wedge \dots \wedge (IZ^{k-1} \equiv IZ^k) \quad \Rightarrow$$

$$X\left(\frac{r_1}{IZ^1}, \frac{r_k}{IZ^k}\right) = X\left(\frac{r_1}{IZ^1}, \frac{r_2}{IZ^2}\right) \rightarrow X\left(\frac{r_1}{IZ^2}, \frac{r_3}{IZ^3}\right) \rightarrow \dots \rightarrow X\left(\frac{r_1}{IZ^{k-1}}, \frac{r_k}{IZ^k}\right) \rightarrow$$

$$X\left(\frac{r_k}{IZ^{k-1}}, \frac{r_{k-1}}{IZ^{k-2}}\right) \rightarrow X\left(\frac{r_k}{IZ^{k-2}}, \frac{r_{k-2}}{IZ^{k-3}}\right) \rightarrow \dots \rightarrow X\left(\frac{r_k}{IZ^2}, \frac{r_2}{IZ^1}\right).$$

As a result of the above operations, only the robots r_i and r_j exchange while all other robots maintain their initial positions. Thus, it is concluded that any pair of robots can exchange their positions and so any robot has access to any IZ of the graph. On the other hand, since according to (ii) all vertices in the graph are located within at least one IZ, the robot can therefore reach any vertex in the graph. This means that the graph is solvable.

Proof of ‘only if’. Now we prove the following statement: “B := If a graph is solvable, then the conditions (i) and (ii) hold” by indirect proof (i.e. proof by contradiction), and for each condition separately:

- i. If we assume that at least two Adjacent Influence Zones in a graph are not Interconnected (i.e., condition (i) revoked), then according to the Theorem 1, exchanging some robots located on two Adjacent—but not Interconnected—IZs would not be possible, and so the graph would not be solvable. This contradicts with the premise of B, and so the condition (i) holds.
- ii. Now we assume that there is a vertex v which does not belong to any Influence Zone (i.e., condition (ii) revoked). In this case, the distance of v to the nearest junction J^p

would be $Dist(v, J^p) \geq H$, which contains $H + 1$ vertices (refer to Definition 4). According to the proof of Lemma 3, in order for a robot on v to move to a vertex connected to J^p , all the vertices between v and J^p (excluding v but including J^p) plus the destination vertex must be empty or able to be emptied (compare to the Fig. 7). Therefore, the total number of required Holes would be

$$H^* = [|Path(v, J^p)| - 1] + 1 = [(H + 1) - 1] + 1 = H + 1 \quad (5)$$

which is in contradiction with the assumption of existing H Holes in the statement B. Thus, it is concluded that any vertex in the graph must belong to at least one IZ, and the condition (ii) holds. \square

A direct ramification of the Theorem 2 is proposed in Corollary 2:

Corollary 2. *The maximum number of robots for which a tree is solvable is $m = |G| - \|S\| - b$, in which $|G|$ is the order of the graph, $\|S\|$ is the length of the Stem, and*

$$b = \begin{cases} 2 & \text{if both ends of the Stem are junctions} \\ 1 & \text{if only one end of the Stem is a junction} \end{cases}$$

Proof. Recall that $H = |G| - m$, in which m is the number of robots on the graph. If we assume that the graph is solvable, then the maximum distances between all its Adjacent junctions cannot exceed $H - 2$. As a result, the length of the longest sequence of Internal Vertices including the two vertices connected to its ends (i.e. the Stem) is bound by $\|S\| \leq H - 2$.

Depending on the types of end-vertices of a Stem, the number of Holes and thus the maximum number of robots in a Solvable Graph is determined as follows:

If the Stem's both end-vertices are junctions, then based on (3):

$$\|S\| \leq H - 2 \Rightarrow \|S\| \leq |G| - m - 2 \Rightarrow m \leq |G| - \|S\| - 2.$$

If the Stem's end-vertices are a junction and a leaf, then based on (1):

$$\|S\| \leq H - 1 \Rightarrow \|S\| \leq |G| - m - 1 \Rightarrow m \leq |G| - \|S\| - 1. \quad \square$$

4. Minimal Solvable Trees

For a specific number of robots (m), a notable subclass of Solvable Trees ST^m is the set of Minimal Solvable Trees (MST^m) which have the minimum number of vertices necessary for accommodating m robots. Considering that the complexity of graph searching operations is directly influenced from the graph size, finding Minimal Solvable Trees would significantly ease the tasks of graph designing and multi robot motion planning. In this section the topologies of Minimal Solvable Trees are introduced. It is noted that for m robots, there can be a number of MSTs with different topologies.

4.1. Minimal stars

As introduced in section 3.1, a $Star^k$ is a tree with k leaves and has a diameter of 2. According to the Lemma 2, in order for a $Star^k$ to be solvable for m robots, it must have at least $H = 2$ Holes. This means that the order of the tree must be

$$|Star^k| = k + 1 \geq m + 2. \quad (6)$$

For verifying this equation we resort to the proof by contradiction: Let's assume that a $Star^m$ is not minimally solvable for m robots. Then a Star with a smaller order (i.e. $|Star^m| < m + 2$) should be minimally solvable. If $|Star^m| \leq m$, then there are no vertices for moving the robots, and so this Star is not solvable. If $|Star^m| = m + 1$, then a robot on a Leaf cannot move to another Leaf since its moves are limited to just one Leaf and the only junction. Therefore, all final configurations are not reachable, and so the graph is not solvable. We conclude that for a Star to be Minimally Solvable for m robots it must have $m + 1$ Leaves, and its order should be

$$|Star^m| = m + 2. \quad (7)$$

4.2 Minimal Extended Stars

An Extended Star (ES^k) has a total of k leaves and internal vertices, all vertices of which belong to the only Influence Zone formed around the only junction. Since according to the Lemma 3 the longest path from the junction (inclusive) must be empty (which is the Stem, S), then $H \geq \|S\| + 1$, or equivalently, $H \geq |S|$. Therefore, the minimum number of vertices of an Extended Star is:

$$|ES^k| = m + |S|. \quad (8)$$

4.3 Minimal General Trees

In a General tree, multiple junctions, and therefore, multiple Influence Zones can exist. According to Theorems 1 and 2, in any solvable tree we have: $Dist(J^p, J^q) \leq H - 2$. The minimum value that H can take to produce a positive distance is $H = 3$. Thus, the order of a Minimal General Tree with at least two junctions is:

$$|G| = m + 3. \quad (9)$$

Any tree not satisfying the conditions of the Theorem 2 is definitely not solvable for m robots, and may be either a non-solvable tree, or a Partially Solvable Tree for m robots (PST^m). Also, regarding the proof of minimality of General Trees, if $m < |T| < m + 3$, then the tree is PST^m . Note that while not being Solvable Trees, PSTs are still solvable for a limited class of problems: those which require only Local Interchanges within either a single Influence Zone, or an Interconnected Influence Zone.

5. Application

Although the results of the previous sections are theoretical in nature, they can be used straightforwardly in real-world applications, such as designing and reshaping transportation networks, railways, traffic roads, AGV routes, and robotic workspaces for multiple moving agents such as trains, vehicles, and robots. On the other hand, in order for a graph to be utilized as the structure of a real network, its topology must be suited to the application it is designed for. Thus, efficiency is an important issue in designing and tailoring applied graphs.

A graph can be considered as efficient if, in addition to being solvable, has no or very few redundant edges and vertices. Lacking redundant elements (i.e. edges and vertices) in a

solvable graph becomes substantially significant particularly when the cost of constructing real world counterparts of graph elements (such as roads, railways, canals, etc.) is remarkably high.

Based on the above discussions, designing an efficient graph encompasses two phases:

- a. *Compatibility* phase, and
- b. *Rightsizing* phase.

In the Compatibility phase, the given graph is made compatible with the needs and constraints of the application. This may include making the graph solvable for the specified number of agents (or robots) by expanding it (if it is not solvable), or reducing the size of the graph if it is solvable for an excessive, unnecessary number of robots. The output of this phase is a Solvable Graph for exactly the predefined number of robots, e.g., m .

In the Rightsizing phase, the Solvable Graph is further examined to find redundant edges or vertices for pruning. We will call the resultant graph a 'Lean' graph since it is fully operational and optimized for the application at hand.

As discussed in Corollary 2, the maximum number of robots for which a graph can be solvable is determined by the order of the graph, $|G|$, the length of the Stem, $\|S\|$ and the type of vertices on the Stem's ends. On the other hand, sometimes it is desirable to modify a given Solvable Graph SG^m in order to accommodate larger or smaller numbers of moving robots. This happens for instance when the graph represents the routes of Automatic Guided Vehicles (AGVs) on plant floor, or railways connecting urban or rural districts.

Converting an SG^m to $SG^{m'}$ has two aspects:

1. If $m' > m$, then the SG^m is Partially Solvable for m' robots (i.e., it is a $PSG^{m'}$). In this case, some vertices and/or edges must be inserted or relocated to produce an $SG^{m'}$.
2. If $m' \leq m$, then the SG^m is solvable for m' robots. In this case, there might be some redundant vertices and/or edges in the graph which can be truncated or relocated to produce a smaller $SG^{m'}$.

Apparently the first case, i.e. the problem of converting an SG^m into an $SG^{m'}$ for $m' > m$, is interesting, in which $n = m' - m$ additional robots should navigate on the graph. Regarding that in an SG^m the maximum number of robots is $m = |G| - \|S\| - b$ ($b = 1$ or 2), accommodating n additional robots requires that the difference $|G| - \|S\|$ be increased by n , or the value of b decreased by 1, if possible.

The graph's expansion or modification is done through four basic operations: Vertex Insertion, Vertex Relocation, Edge Insertion, and Edge Relocation. It should be noted that increasing the number of robots by decreasing the value of b is possible only via Edge Insertion and Edge Relocation operations.

In expanding the size of the graph in the Compatibility phase special care must be taken to comply with the limitations and conditions of the application. For example, there might be no sufficient space for inserting a vertex or edge, or inserting an edge could be economically or technically infeasible. Whenever the graph is not capable of accommodating the specified number of agents, a possible engineering solution could be either reducing the number of agents or inserting the required graph elements such that the sustained expense is minimized.

On the other hand, in reducing/pruning the graph, some vertices might be indelible as they are essential to the system, as the positions for loading, feeding, or parking machines or vehicles. Therefore, possible operations in the Rightsizing phase are Vertex Deletion and Edge Deletion operations, and Vertex/Edge Relocation and Insertion are not considered.

It must be noted that although deletion of redundant vertices or edges from the graph may reduce the overall cost of network design and construction, it may lead to higher costs of agents' movements as well, since by possessing a more limited space for manoeuvring and reshuffling, agents may be forced to travel/stop more frequently for resolving deadlocks.

5.1 An example

Now a practical illustrative example is provided for demonstrating a typical application of the results of the previous sections. Specifically, it is shown how designing a Solvable Graph and then optimizing it might help solving an industrial shop floor problem.

Suppose that a factory is consisted of a workshop and two warehouses, one for raw material and one for finished products. The production area accommodates 3 milling, 2 turning, 1 drilling, and 1 grinding machines, as well as a heat treatment line and an assembly station. Each of these machines or departments requires certain amount of raw material (or unfinished parts) to be loaded, processed, unloaded and then transported to other divisions of the factory. The layout of the factory and the loading/unloading positions for each machine and department are shown in Fig. 8.

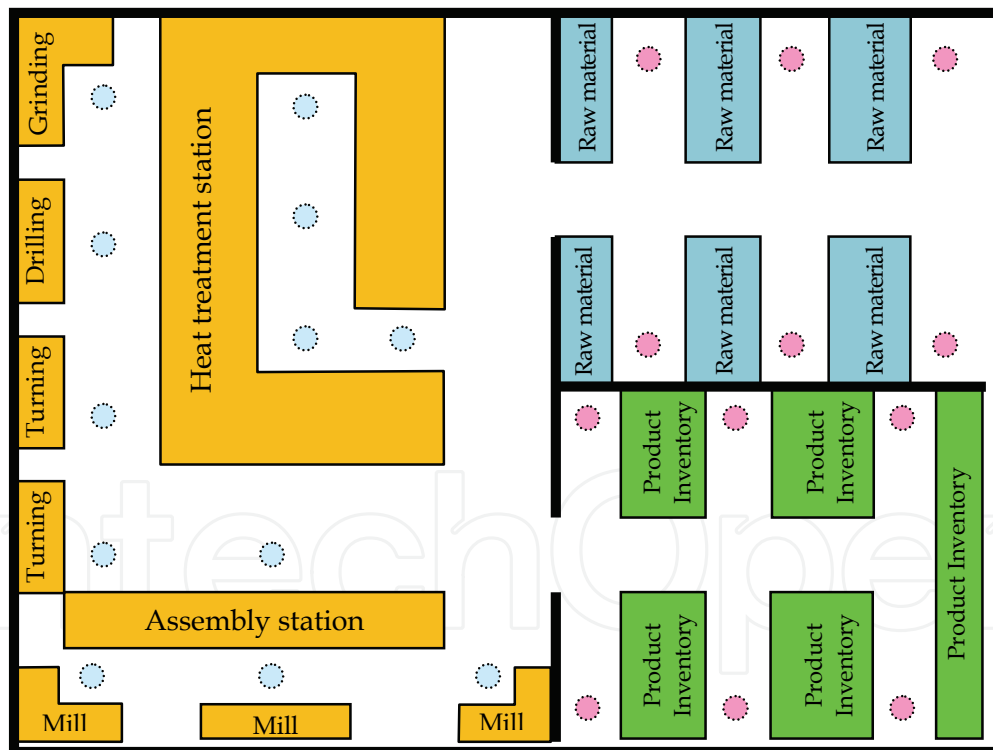


Fig. 8. Layout of the factory. The specific locations for loading/unloading parts and material are shown with small circles

The materials and products are currently carried manually by carts and pallets. However, the management decides to upgrade the factory's transportation system and utilize

Automatic Guided Vehicles (AGVs). However, there are a number of constraints and conditions in this regard.

Constraints:

1. The AGVs can only move along predefined paths realized by embedding special type of wires in the floor and covering them with epoxy resins for safety and durability.
2. Due to space limitation, no two AGVs can move along a single passageway or corridor side by side, and should therefore wait until the passageway (i.e. the wire track) is evacuated and emptied from other AGVs.
3. AGVs can only stop at certain locations for loading/unloading or changing their tracks. They cannot stop at any point on the track other than the predefined spots.
4. Since the factory's production system is Job-Shop, the volume of interdepartmental transportation is high. Therefore, all AGVs need to have access to all points of the transportation network. In other words, all AGVs must be able to reach any destination from any initial position.
5. Each AGV is equipped with some batteries that actuate it and provide power for its sensors and control unit, and therefore needs to be recharged daily. The recharging is done during the factory's idle times, i.e. at nights. However, for safety reasons, recharging hubs are located in warehouses so that AGVs can be recharged and kept secured at nights.
6. The cost of laying out wire tracks is proportionate to their lengths. Also, situating each stop point requires additional expense due to the special sensors and circuits needed. These initial costs are much higher than the costs of increased movements the AGVs need to make for avoiding collisions or deadlocks.
7. The management has decided to mechanise the material handling system through two phases: in the first phase, 10 AGVs are planned to operate in the factory, and in the second phase, in parallel with increasing the production volume and variety, the number of AGVs will be increased to 15.

Problem:

The problem is to design a network of wire tracks and stopping stations such that the total cost of upgrading is minimized while the above constraints are fulfilled. In other words, it is desired to plan a Solvable Graph for 10 AGVs (for the first phase) which is efficient (minimal) in terms of the number of vertices and edges.

Solution:

As a starting point, the Voronoi Diagram of the factory is calculated, which is the collection of points equidistant from two or more facilities or walls (refer to Section 1 for more details). Implementing the Voronoi Diagram is quite proper since it yields the safest paths passing through narrow corridors and production facilities. Fig. 9 depicts the Voronoi Diagram of the factory. As can be seen in the Fig. 9, the Voronoi diagram has some small redundant edges connected to concave corners of the workspace. These edges, however, can be easily pruned (removed) from the diagram. Also, there are some vertices very close to each other (as shown in the dashed circle) which can be merged for simplicity and practicality.

The next step is to modify the pruned Voronoi Diagram to comply with the requirements of loading/unloading stations imposed by the constraint 3. This is done by superimposing the Voronoi Diagram and the stations' locations, as illustrated in Fig. 10.

Nearly all stopping stations situate at proper positions on the Voronoi Diagram, except for the lower-left corner, for which we proposed a simplifying solution, as depicted in Fig. 11.

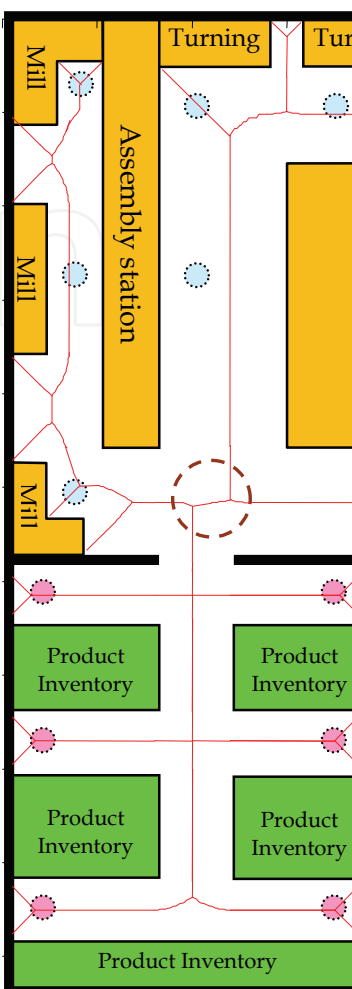


Fig. 9. Voronoi Diagram of the factory, which is the collection of safest paths amid facilities.

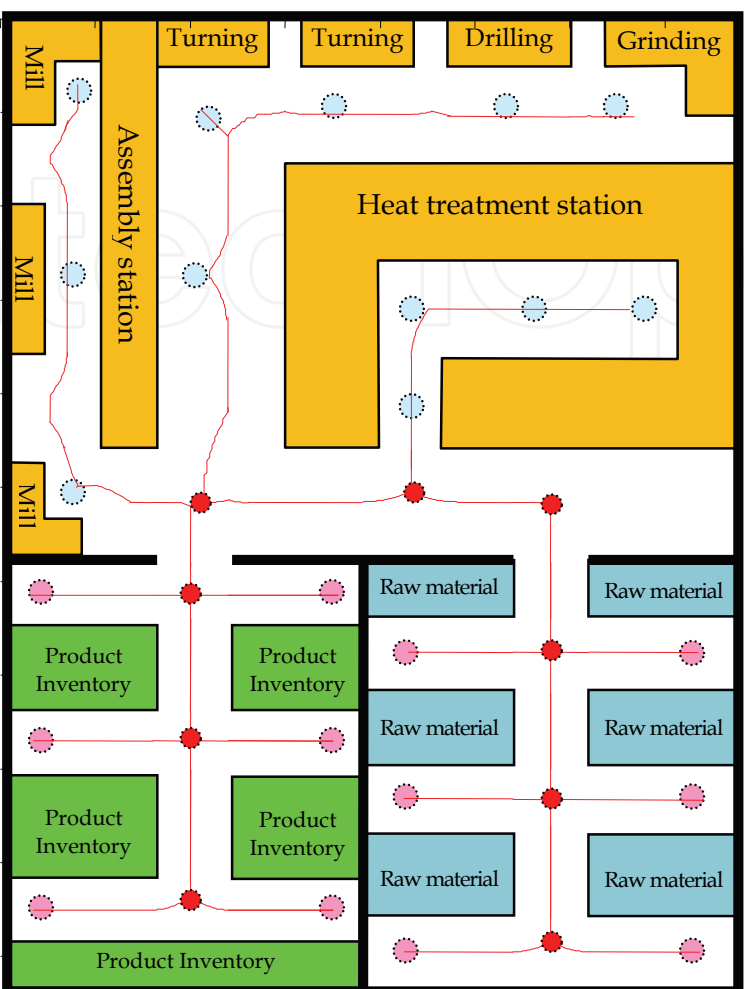


Fig. 10. Superimposition of the pruned Voronoi Diagram and the stopping stations

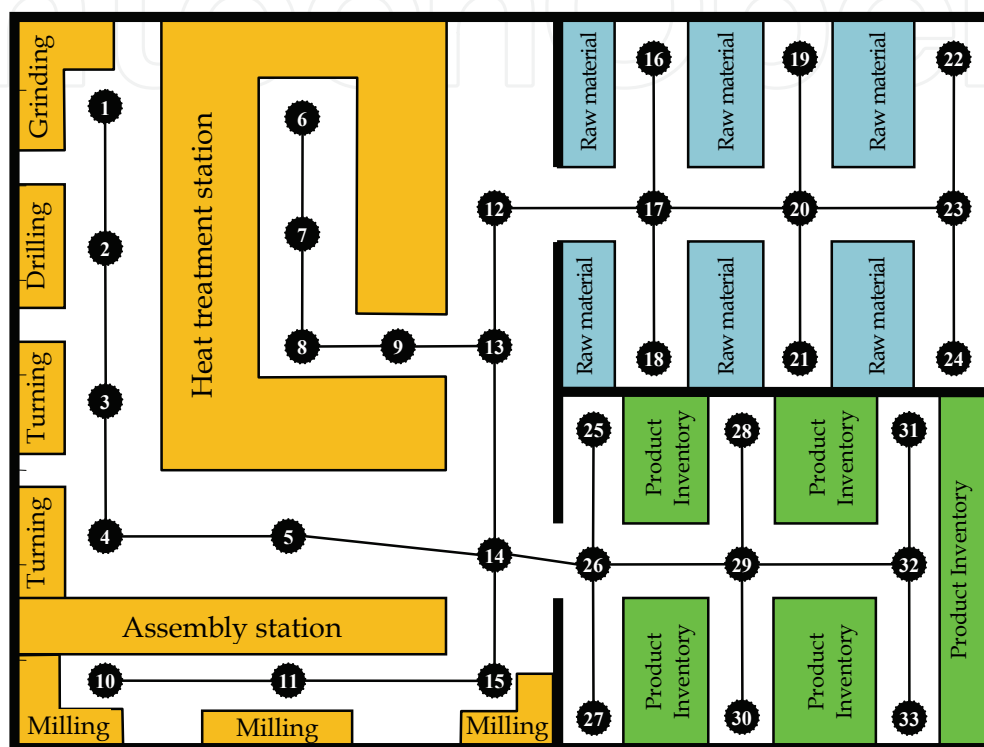


Fig. 11. A simplified AGV transportation network for the factory

In the Fig. 11 there are 8 junctions and the vertices v_1, v_2, v_3, v_4, v_5 and v_{14} constitute the Stem. As stated in the constraint 7 above, the management has decided to operate the network for 10 AGVs in the first phase. By taking into consideration all possible vertices of the network and according to the Corollary 2, the maximum number of AGVs that can travel in the whole factory is $m = |G| - \|S\| - 1 = 33 - 5 - 1 = 27$. Although this implies that a large number of AGVs can operate, they have to make use of all vertices (including those in warehouses) for accessing all stations, which may make their travels long and unnecessary, and will cause disorder and insecurity in the warehouses.

The engineering team responsible for designing the network recommends limiting the scope of the AGVs' motions to the production workshop, such that no AGV can enter the warehouses except for material loading or unloading, and therefore is only allowed to travel in the production area for resolving deadlocks with other AGVs. In this case, the maximum number of AGVs able to move will be $m = 15 - 5 - 1 = 9$. Since this number is less than the required number of AGVs the management has planned, the vertex v_{26} , which is the nearest

vertex in the warehouses to the production area, is selected to be included in the operational network. This resolution will not affect the size of the network's Stem, and so the number of operational AGVs will increase to $m = 16 - 5 - 1 = 10$, which will make the network compatible with the requirements of the management.

In order to increase the production volume and variety, the number of AGVs is planned to increase to 15. For this phase, two possible scenarios among others are:

1. By utilizing 5 additional vertices located in the warehouse areas, such as v_{17} , v_{20} , v_{23} , v_{29} and v_{32} , the maximum number of AGVs will increase to $m = (16 + 5) - 5 - 1 = 15$.
2. By merging the vertices v_8 and v_9 , and converting the vertex v_3 into a junction, the Stem size will be $\|S\| = 3$. Moreover, by utilizing 3 additional warehouse vertices such as v_{17} , v_{20} and v_{29} , the number of operational AGVs will increase to $m = (16 + 3) - 3 - 1 = 15$.

Evidently, each of the above solutions should undergo financial and technical feasibility studies to be selected and finalized.

6. Concluding remarks

The time complexity of the proposed method for verifying the solvability of a graph without explicitly solving it is in $O(n)$ which is spent on identifying the Stem, where n is the number of graph vertices. Also, calculating the maximum number of robots operable on a given graph can be performed in the same time order.

In contrast, investigating the solvability of a Multi Robot Motion Planning problem of m robots on a graph with n vertices through exhaustive enumeration will require

$$p = \frac{n!}{(n-m)!} \quad (10)$$

different permutations of robots to be checked, which is far beyond the linear time order of the presented algorithm.

Moreover, verifying whether a graph is SG^m would require

$$\sum_{i=1}^m \left(\frac{n!}{(n-i)!} \right)^2 \quad (11)$$

operations to be checked, for any initial and final configurations, which is again exponentially time consuming. These figures demonstrate the effectiveness of our findings in terms of required time and memory.

In designing transportation networks for multiple autonomous agents (such as mobile robots, AGVs, cars, etc.) which can merely move along the network's arcs, it is important to make sure that the graph has a proper topology and sufficient number of vertices (relative to the number of agents) to enable the agents' motion planning. This chapter dealt with the topology of Solvable Graphs and introduced the new concept of Minimal Solvable Graphs

and investigated their properties, which are the smallest graphs that satisfy the feasibility conditions for multi robot motion planning for any initial and final configurations of robots. Finding Minimal Solvable Trees will significantly ease the motion planning task for multiple robots on trees.

In this chapter we assumed that the robots move sequentially on internal vertices and leaves of the tree. For further research it is possible to assume all moves to be concurrent and find the minimum sequence of moves to reach the final configuration with different edge lengths and robots velocities. Also, further research is underway for working out solvability conditions for Cyclic Graphs and Partially Solvable Graphs.

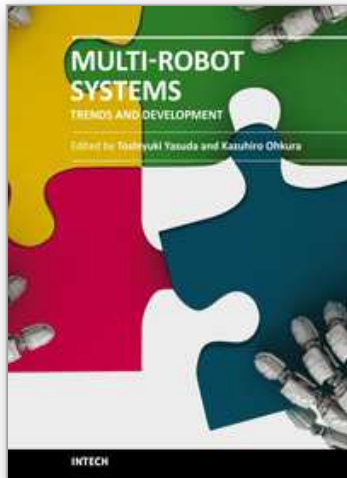
7. References

- Auletta, V.; Monti, A.; Parente, D. & Persiano, G. (1999). A Linear-Time Algorithm for the Feasibility of Pebble Motion on Trees, *Algorithmica*, Vol. 23, No. 3, pp. 223-245, ISSN 0178-4617.
- Bennewitz M.; Burgard, W. & Thrun, S. (2001). Optimizing schedules for prioritized path planning of multi-robot systems, *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 271-276, ISBN 0-7803-6576-3, Seoul, Korea, May 2001, IEEE, New Jersey.
- Canny, J. F. (1988). *The Complexity of Robot Motion Planning*, The MIT Press, ISBN 978-0-262-03136-3, Cambridge, Mass.
- Choset, H.; Lynch, K. M.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. E. & Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press, ISBN 978-0-262-03327-5, Boston.
- Diestel, R. (2000). *Graph Theory*, Springer-Verlag, ISBN 978-3-642-14278-9, Heidelberg.
- Hwang, Y. K. & Ahuja, N. (1992). Gross motion planning - A survey, *ACM Computing Surveys*, Vol. 24, No. 3, pp. 219-291, ISSN 0360-0300.
- Kant, K. & Zucker, S. W. (1986). Toward efficient trajectory planning: the path-velocity decomposition, *International Journal of Robotic Research*, Vol. 5, No. 3, pp. 72-89, ISSN 0278-3649.
- Kornhauser, D.; Miller G. & Spirakis, P. (1984). Coordinating pebble motion on graphs, the diameter of permutations groups and applications, *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, pp. 241-250, ISBN 0-8186-0591-X, Singer Island, FL, October 1984, IEEE, New Jersey.
- Latombe, J. C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, ISBN 978-0792392064, London.
- Masehian, E. & Hassan Nejad, A. (2009). Solvability of multi robot motion planning problems on trees, *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pp. 5936-5941, ISBN 978-1-4244-3803-7, St. Louis, USA, October 2009, IEEE, New Jersey.
- Onn, S. & Tennenholtz, M. (1997). Determination of social laws for agent mobilization, *Artificial Intelligence*, Vol. 95, 1997, pp. 155-167, ISSN 0004-3702.
- Ryan, M. R. K. (2008). Exploiting subgraph structure in multi-robot path planning, *Journal of Artificial Intelligent Research*, Vol. 31, No. 1, pp. 497-542, ISSN 1076-9757.

Wilson, R. M. (1974). Graph puzzles, homotopy, and the alternating group, *Journal of Combinatorial Theory, Series B*, Vol. 16, pp. 86-94, ISSN 0095-8956.

IntechOpen

IntechOpen



Multi-Robot Systems, Trends and Development

Edited by Dr Toshiyuki Yasuda

ISBN 978-953-307-425-2

Hard cover, 586 pages

Publisher InTech

Published online 30, January, 2011

Published in print edition January, 2011

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

How to reference

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

Ellips Masehian and Azadeh H. Nejad (2011). Graph-Based Multi Robot Motion Planning: Feasibility and Structural Properties, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/graph-based-multi-robot-motion-planning-feasibility-and-structural-properties>

INTECH

open science | open minds

InTech Europe

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

InTech China

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

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen