

Solvability of Multi Robot Motion Planning Problems on Trees

Ellips Masehian and Azadeh H. Nejad

Abstract—Multi robot motion planning problems can be solved very efficiently when the Configuration Space is mapped onto a Graph. Before planning, however, it must be assured that the constructed graph is reachable (solvable) for the given number and configuration of robots. Solvable Trees are types of trees that let any arrangement of a specified number of robots be reached from any initial arrangement through sequential moves of robots on vertices of the tree. In this paper the properties of Solvable Trees are investigated, and Minimal Solvable Trees, which are the smallest solvable trees in terms of the number of vertices, are introduced as a new concept. Also, a new algorithm with linear time complexity is proposed for deciding whether a multi robot motion planning problem has a solution on a tree, without explicitly solving it.

I. INTRODUCTION

THE general motion planning problem is known to be PSPACE-hard and NP-complete. Most of the solution methods for robot motion planning are variations of a few general approaches: Roadmap, Cell Decomposition, Potential Fields, mathematical programming, and heuristic methods, which are broadly surveyed in [1].

Roadmap approaches reduce the workspace into a graph with nodes including the starts and goals of the robot, and find a collision-free path through graph search. Although classic approaches like Visibility Graph, Voronoi Diagram, and Cell Decomposition have proved to be effective for single-robot problems, they do not provide straightforward solutions to Multi Robot Motion Planning (MRMP) problems.

Space is the most limiting constraint in a typical MRMP problem: often, because of lack of sufficient space around robots, they cannot reach their goals 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 robots cannot pass by each other. To resolve a deadlock, robots should reshuffle, circumnavigate, detour, or regulate their speed (also known as Velocity Tuning). Another approach is the Prioritized Planning, in which the robots move according to a priority scheme, as discussed in [2], [3], and [4].

By reducing the workspace into a graph with nodes including the starts and goals of all robots, the multi robot motion planning problem then turns 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 the obstacles existing in the workspace. Another benefit of

graph-based MRMP is its reduced computational time.

The main question in designing a predefined graph is to find out whether the graph is 'reachable' (solvable) for any initial and final configurations. A Configuration is an arrangement of robots on the vertices of the graph such that no vertex is occupied by more than one robot. Solvable graphs allow the transition of any initial configuration of robots to a final state via their sequential moves.

The first work trying to find a relation between a graph's topology and the number of 'beans' moving on it is due to Wilson, in which only bi-connected graphs are studied, and $k = n - 1$ (k is the number of beans, n the number of vertices) [5]. This study is improved in [6] by generalizing the decision problem for all graphs and for any number of robots. In [7] the above problem is studied as pebble motion problem by generalizing the 15-puzzle, and a linear algorithm for deciding the reachability of trees is presented. However, both [6] and [7] only cover cases where the set of empty nodes remain unchanged in start and goal configurations. In [8] the possibility of reaching destinations of connected subgraphs is studied, by simplifying the MRMP between some predefined subgraphs named stacks, halls, rings, and cliques.

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. 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 maneuverability 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 paper, 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 II, in section III the conditions for a tree to be solvable are investigated and the maximum number of robots located on it is determined. Two new concepts, MSTs and PSTs are introduced in Section IV. In Section V we propose an algorithm for deciding whether an MRMP with a given tree and initial and final configurations has a solution, without explicitly solving it.

Authors are with the Faculty of Engineering, Tarbiat Modares University, Tehran, Iran. Corresponding author's e-mail: masehian@modares.ac.ir.

II. DEFINITIONS

Before we proceed, let us present three new fundamental and correlated concepts:

- A *Solvable Tree* is a tree on which any configuration of at most m robots can be reached from any initial configuration through their moves on tree edges, and is shown by ST^m .
- A *Partially Solvable Tree* is a tree on which only *some* configurations of m robots can be reached from any initial configuration, and is shown by PST^m .
- A *Minimal Solvable Tree* is an ST^m having the least possible number of vertices, and is shown by MST^m .

In addition to the standard terminology used in Graph Theory, there are some new definitions and symbols defined specifically for this work, presented in Table I. Also, some concepts are shown graphically in Fig. 1(a).

Here we review some basic concepts in Graph Theory [9]: Let $T = (V, E)$ be a tree, with the set of vertices V and set of edges E . The number of vertices also indicates the *order* of T , denoted by $|T|$. The number of vertices connected to the vertex v is called its *Degree*, $d(v)$. *Leaves* of a tree, $L(T)$, are vertices with $d(v) = 1$, and *Internal Vertices* of a tree, $I(T)$, are vertices with $d(v) > 1$. $Dist(v, u)$ is defined as the length of the shortest path between two vertices v and u .

TABLE I
DEFINITIONS OF USED TERMS AND SYMBOLS

Term / Symbol	Description
$O \in \{v \mid d(v) \geq d(u), \forall u \in V\}$	<i>Origin</i> : The vertex with maximum degree in the Tree. If not unique, selected such that the maximum Level of all vertices is kept minimal.
$l(v) = Dist(v, O)$	<i>Level</i> : The minimal distance of vertex v from O along the edges of the Tree.
$J = \{v \mid v \text{ is MI}, d(v) \geq 3\}$	<i>Junctions</i> : The set of Internal Vertices with degrees of more than 2. It may be O as well.
$H(S)$	<i>Holes</i> : The set of empty vertices not occupied by robots at initial configuration in the subtree S . The size of $H(S)$ is denoted by $h(S) = H(S) $, and when S is the whole Tree, we set $H = h(T)$.

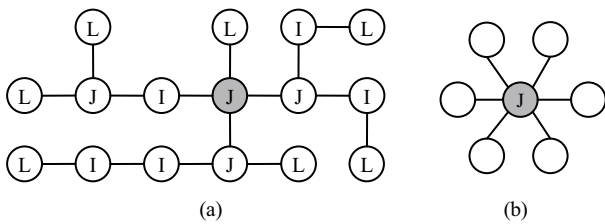


Fig. 1. (a) Some concepts: L = Leaf, I = Internal Vertex, and J = Junction. The central gray junction is selected the Origin of the tree. (b) A 6-Leaf Star.

III. SOLVABILITY OF 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.

On the other hand, Leaves and Junctions' branches can be used for robots' maneuvers 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.

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.

A. Star Trees

A Star is a complete bipartite graph with only one vertex in one part and one or more vertices in the other part. Stars are trees with diameters equal to 2, and have only one Junction. Fig. 1(b) illustrates a 6-Leaf Star.

Here we propose a lemma for the solvability of Stars:

Lemma 1. *A Star with $m+1$ Leaves is Solvable for m robots.*

Proof. A Star with $m+1$ Leaves has order of $m+2$ (e.g. Fig. 1(b) for $m = 5$), and two of its vertices are not occupied by robots. Depending on the robots' initial configuration, these two vertices are called *Holes* and may either be both Leaves (case *a*), or a Leaf and the only Internal Vertex (case *b*).

To prove the solvability of Star, various possibilities of initial and final configurations of robots can be considered in the form of four Scenarios enlisted in Table II.

TABLE II
POSSIBLE SCENARIOS FOR ROBOTS' CONFIGURATIONS ON STARS

	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 internal vertex
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 internal vertex	Scenario 3	Scenario 4

Scenario 1: In this scenario a Leaf and the Internal Vertex are initially empty (case *b*). 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 *b* holds again. The robots should occupy their goals in the same manner as in the Scenario 1, with the consideration that occupying the internal vertex should be the last move. The Star is therefore solvable.

Scenario 3: In this scenario, initially one robot is on the Star's internal vertex and $m-1$ robots are on the Leaves, and so the case *a* holds true. Through a single move of the robot located on the internal vertex 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 *a* applies for this scenario. Again by moving the robot on the internal vertex to an empty Leaf, this scenario converts to the Scenario 2, and the problem can be solved accordingly. \square

The concept of Star trees can be generalized as follows:

An *Extended Star* is a tree with one Junction (hence the Origin) and more than one Internal Vertices. In other words, Extended Stars can be constructed by adding new Leaves to a Star, such that no other Junction is formed.

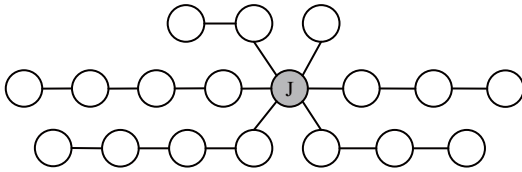


Fig. 2. An Extended Star with a maximum Level $l_{\max} = 4$.

The following Lemma deals with the solvability of Extended Stars:

Lemma 2. *The maximum number of robots for which an Extended Star is solvable is $m = |T| - l_{\max} - 1$, in which l_{\max} is the maximum Level in the tree.*

Proof. The maximum distance from the Extended Star's Junction J to any vertex in the tree is $l_{\max} = \max\{l(v), \forall v \in V\}$. It is obvious that in order for the tree to be solvable, any two robots must be able to interchange their positions. The worst case of interchanging occurs when a robot r located on the vertex with $l(v) = l_{\max}$ has to move to the depth of another branch of the tree. Since there is only one Junction in the tree, the robot should pass through the Junction, and so at least the path P connecting vertex $v(r)$ and J , plus another vertex connected to the Junction (i.e. not on P) must be either initially empty, or able to be emptied by motions of other robots. This means that at least $l_{\max} + 1$ vertices in the tree should be empty. That is, $H = l_{\max} + 1$, and so the maximum number of robots will be $m = |T| - H = |T| - l_{\max} - 1$. \square

B. General Trees

General Trees are trees with multiple Junctions. A general tree can be partitioned into a number of zones formed around each Junction. As a result, the solvability of a general tree can then be assessed in two phases: first, the possibility of robots' maneuvers *within* each zone is checked (i.e., local interchanges), and second, the possibility of robots' interchanges *among* all zones is verified (i.e., global interchanges).

Partitioning of a general tree is based on the number of holes (empty vertices) calculated by $H = |T| - m$, where $|T|$ is the order of the tree (or the number of its vertices).

The conditions for local (micro level) and global (macro level) interchangeabilities in general solvable trees are discussed in Lemmas 3 and 4. Before that, some definitions are provided below:

Definition 1. Any two Junctions are considered *Near* if there is no other Junction between them. That is,

$$Near(j_p, j_q) = \begin{cases} 1 & \text{if } \nexists j_r \in \text{Path}(j_p, j_q), j_r \in J \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\text{Path}(j_p, j_q)$ is the set of vertices connecting j_p and j_q .

Definition 2. The set of vertices around a Junction satisfying the following condition is called the *Influence Zone* (IZ) of that Junction. That is,

$$IZ_k = \{v \mid Dist(j_k, v) \leq H - 1, \forall v \in V\} \quad (2)$$

Two IZs are considered *Near* if their Junctions are *Near*. Fig. 3 shows Influence Zones for all four Junctions of a tree.

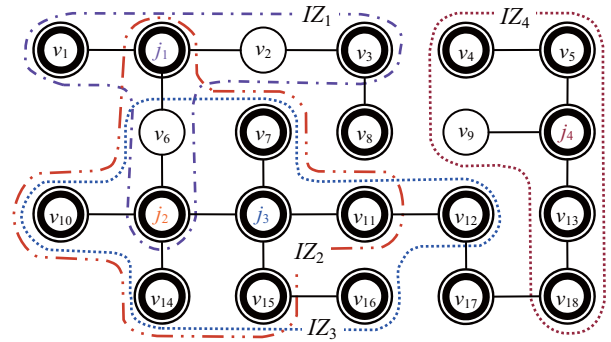


Fig. 3. Influence Zones (IZs) of the Junctions of a sample tree with $H = 3$. Note that v_8 and v_{17} do not belong to any IZ.

Definition 3. An *Interconnected Influence Zone* (IIZ) is the set of all Influence Zones (IZs) such that for any two *Near* IZs in the set, the distance between their Junctions is not more than $(H - 2)$. Mathematically,

$$IIZ_t = \{IZ_p \mid Dist(j_p, j_q) \leq H - 2\} \quad (3)$$

for all $Near(IZ_p, IZ_q) = 1$ and $\{j_p, j_q\} \in J$. For example, in Fig. 3, IZ_2 and IZ_3 merge and form IIZ_1 (see Table III). Note that once interconnected IZs merge and form a new IIZ, they are omitted from the list of existing IZs. Even so, the remained IZs are still labeled according to their Junction number.

C. Condition for Local Interchanges

Local robot interchanges are considered to be any type of rearrangements from an initial to a final configuration taken place inside an Influence Zone of a Junction. The following lemma proposes the condition which an IZ must have to be solvable.

Lemma 3. *Any Influence Zone having H Holes is solvable.*

Proof. Based on definitions of Extended Star and Influence Zone, it is inferred that an Extended Star is a special subset of Influence Zone, having only one Junction. Regarding the proof of Lemma 2 and that Junctions are vertices at which robots can change their course of motion or exchange positions, the solvability condition in Extended Stars requires at least more empty vertices than in IZs. In other words, a lower limit for the number of holes in an IZ is determined from the number of necessary holes in an Extended Star with the same depth of branches (which is $H = l_{\max} + 1$). Since by definition, in an IZ, $l_{\max} = H - 1$, then the number of holes is at least $H^* = l_{\max} + 1 = H - 1 + 1 = H$. \square

For example in the Fig. 3 if all the holes were on $\{v_6, j_1, j_2\}$ (situated in IZ_1 and IZ_2), then the robots on $\{v_1, v_2, v_3\}$ and $\{v_7, v_{10}, v_{11}, v_{14}, v_{15}, j_3\}$ could interchange in IZ_1 and IZ_2 , respectively.

D. Conditions for Global Interchanges

In this stage the conditions for interchangeability of any two robot located in *Near* Influence Zones are investigated. Obviously, the results can be generalized to the whole tree due to the transitive property of sets (in this case, IZs). The following lemma provides conditions for global interchanges of robots, hence the solvability of the tree.

Lemma 4. Any General Tree having H Holes is solvable iff the following conditions are satisfied:

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

Proof. (a) We first assume that a tree with H Holes is solvable if the conditions (i) and (ii) hold, and prove it by direct proof: 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 can not only move to anywhere in its IZ, but also to any vertex in its Near interconnected IZ (due to (i) and Lemma 3). Furthermore, it can also move to any vertex in any IZ in the tree since by (i) all near IZs are Interconnected, having distances less than or equal to $H - 2$ between their Junctions. On the other hand, since according to (ii) all vertices in the tree are located within at least one IZ, the robot can therefore access to any vertex in the tree. This means that the tree is solvable. \square

(b) Now we assume that in a tree with H Holes the conditions (i) and (ii) hold if the tree is solvable. We will prove it by indirect proof (i.e. proof by contradiction) for each condition separately:

Case (b.i): Suppose that the distance of two Near Junctions (i.e. the number of edges between) is greater than $H - 2$. Take an arbitrary robot on any vertex: in order for this robot to move from its IZ to any vertex in its Near IZ, the path connecting their two Junctions (inclusive) and the destination vertex must be empty. Regarding that by the current counterargument $Dist(j_p, j_q) > H - 2$, then it can be at least $Dist(j_p, j_q) = H - 1$, which contains one more vertex. Therefore, counting the destination vertex, the total number of required Holes in the tree would be $H^* = [Dist(j_p, j_q) + 1] + 1 = [(H - 1) + 1] + 1 = H + 1$ (4) which is in contradiction with the assumption of existing H Holes in (b). Thus, it is concluded that $Dist(j_p, j_q) \leq H - 2$, and the condition (i) holds. \square

Case (b.ii): we assume that there exists at least one vertex v that does not belong to any IZ. According to the definition of IZs, the distance of v to its nearest Junction j_p is at least H (i.e. $Dist(v, j_p) = H$), which contains $H + 1$ vertices. In order for a robot on v to move to a vertex connected to j_p , the path connecting v and j_p and the destination vertex must be empty. So the total number of required Holes would be $H^* = [Dist(j_p, j_q) + 1] + 1 = [(H - 1) + 1] + 1 = H + 1$ (5) which is in contradiction with the assumption of existing H Holes in (b). Thus, it is concluded that any vertex in the tree must belong to at least one IZ, and the condition (ii) holds. \square

For example in Fig. 3, if the robot on v_{10} wants to move to v_1 , in addition to the vertices on its way (i.e. j_2, v_6, j_1), it needs v_1 to be empty; which is not so, and the move is impossible.

IV. MINIMAL AND PARTIALLY SOLVABLE TREES

A Minimal Solvable Tree (MST^m) is the smallest tree on which any configuration of at most m robots can be reached from any initial configuration through their moves on the tree's edges, and is shown by MST^m . In this definition, 'smallest' is expressed and measured in terms of the number

of vertices. It is noted that for a specific number of robots (m), there can be some MST^m with different topologies.

Minimality of Stars: For proving that a Star with $m+1$ Leaves is an MST^m , let's assume that the Star with $m+1$ leaves is not minimally solvable for m robots. Then a star with a smaller order (i.e. $|T| < m+2$) should be minimally solvable. If $|T| \leq m$, then there are no vertices for moving the robots, and so this Star is not solvable. If $|T| = m+1$, then a robot on a Leaf cannot move to another Leaf since its moves are limited to just one Leaf and the only Internal Vertex. Therefore, all final configurations are not reachable, and so the graph is not solvable. We conclude that for a Star to be Minimally Solvable it must have an order of $m+2$ (i.e. should have $m+1$ Leaves). \square

Minimality of Extended Stars: For Extended Stars, all vertices belong to the only IZ. Regarding that the longest path from (and including) the Junction must be empty, $H = l_{\max} + 1$, and therefore $|T| = m + H$. \square

Minimality of General Trees: Regarding that $Dist(j_p, j_q) \leq H - 2$, the minimum value that H can take to produce a positive distance is 3. Thus the order of an MST^m is $|T| = m + H = m + 3$, and the order of an ST^m is $|T| > m + 3$. \square

Any tree not satisfying the conditions of the Lemma 4 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. The next Section presents an algorithm for determining whether an MRMP problem is solvable on a given PST or not.

V. DETERMINING THE SOLVABILITY OF A PROBLEM

An important issue in multi-robot routing problems is deciding whether a final configuration of robots on a tree is reachable from an initial configuration through sequential moves of robots. This problem is addressed in works such as [6], [7], and [10], though none have dealt with general cases.

Any MRMP problem for m robots is feasible on ST^m and MST^m , regardless of the initial and final configurations of robots. On PSTs, however, only some problems have solution. In this Section an algorithm is presented for determining the feasibility of a problem on a given tree, regardless of its type, without explicitly solving it. This algorithm is useful for verifying the existence of a solution when the tree is PST, or there is no prior information about its solvability.

Before explaining the algorithm, two definitions are presented: Any vertex in a Zone (IZ or IIZ) having a distance of $H - 1$ to the nearest Junction of that Zone is called a *Port* of that Zone, and is shown by $Pt(IZ_p)^i$ or $Pt(IIZ_i)^i$, respectively for IZ_p and IIZ_i . Also, the subtree connected to a port i of a Zone is called the *Linked Subtree* of that Port, and is shown by $LS(Pt(IZ_p)^i)$ or $LS(Pt(IIZ_i)^i)$, respectively for IZ_p and IIZ_i . The above concepts are explained in Table III for the example in Fig. 3.

TABLE III
IZS, IIZS, PORTS AND LINKED SUBTREES OF THE SAMPLE TREE

IZs	Ports	Linked Subtrees
$IZ_1 = \{j_1, v_1, v_2, v_3, v_6, j_2\}$	$Pt(IZ_1)^1 = j_2$ $Pt(IZ_1)^2 = v_3$	$LS(Pt(IZ_1)^1) = V \setminus \{IZ_1, v_8\}$ $LS(Pt(IZ_1)^2) = v_8$
$IZ_4 = \{j_4, v_4, v_5, v_9, v_{13}, v_{18}\}$	$Pt(IZ_4)^1 = v_{18}$	$LS(Pt(IZ_4)^1) = V \setminus IZ_4$
IIZs	Ports	Linked Subtrees
$IIZ_1 = \{j_1, j_2, j_3, v_6, v_7, v_{10}, v_{11}, v_{12}, v_{14}, v_{15}, v_{16}\}$	$Pt(IIZ_1)^1 = j_1$ $Pt(IIZ_1)^2 = v_{12}$	$LS(Pt(IIZ_1)^1) = \{v_1, v_2, v_3, v_8\}$ $LS(Pt(IIZ_1)^2) = IZ_4 \cup v_{17}$

The algorithm is based on a key concept, the ‘Maximum Reachability Space’ (MRS_i) of robot r_i , which is defined as a subtree that is accessible for r_i .

The outline of the algorithm is as follows: After determining IZs and IIZs, the MRS of each robot is calculated. If the goals of *all* robots lie inside their MRS, then it is checked if the robots intercept each others’ start-to-goal moves. If no blockings occur, then the problem is solvable. On the other hand, if the goal of at least one robot does not lie in its MRS, or the goal is not accessible because of interceptions, then the problem has no solution. The algorithm’s details are given in three phases as follows:

A. Initialization

Step 1. Determine the following parameters and sets:

- Start and Goal positions (s_i and g_i) and shortest path (SP_i) of each robot r_i ,
- The number of Holes (H), and
- All Zones (i.e. IZs and IIZs),
- Ports of all Zones (i.e., $Pt(IZ_p)^i$ and $Pt(IIZ_t)^i$), $\forall p, t, i$,
- Linked Subtrees of all Ports of all Zones (i.e., $LS(Pt(IZ_p)^i)$ and $LS(Pt(IIZ_t)^i)$), $\forall p, t, i$.

B. Maximum Reachability Space Calculation

Step 2. Calculate the MRS_i for each robot according to the procedure described in Fig. 4.

Step 3. If for any robot r_i its goal g_i is not in MRS_i then the problem has no solution. Otherwise, go to the Deadlock Situations Checking phase.

C. Deadlock Situations Checking

In this phase the shortest start-to-goal paths are calculated independently for each robot. If some paths intersect at certain vertices, then some robots may block other robots’ moves to their goals, which are known as *deadlocks*.

All possible cases of deadlocks occurred for two robots moving at same or opposite directions in a tree are illustrated in Table IV. The relative directions of each pair of robots are determined based on the order of vertices visited by each robot. For instance in Case 1, $SP_i = \{2 \rightarrow 3 \rightarrow 4\}$ and $SP_j = \{1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$, and so $SP_i \subset SP_j$. In Case 3, $SP_i = \{5 \rightarrow 4 \rightarrow 3 \rightarrow 2\}$ and $SP_j = \{1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$, and so $SP_i \subset \neg SP_j \wedge s_i = g_j$. The symbol ‘ \neg ’ indicates the path in reverse order.

Step 4. Check the deadlock situations of all robots whose MRS’s are different. If any of the deadlocks in Table IV occurs then the problem is not solvable. Otherwise, there is a solution to the given MRMP problem.

Procedure MRS

```

1 for each robot  $r_i$  do
2   if  $H = 1$  then go to 25.
3   Find the set of Zones (IZ or IIZ) that contain the start ( $s_i$ ) of  $r_i$  (call  $\check{Z}_s$ ).
4   if  $\check{Z}_s = \emptyset$  then go to 25.
5   Find the set of Zones that contain all the vertices of  $SP_i$  (call  $\check{Z}_p$ ).
6   if  $\check{Z}_p = \emptyset$  then  $\check{Z}_p \leftarrow \check{Z}_s$ 
7   for each Zone in  $\check{Z}_p$  (call  $Z_p^j$ ) do
8     Find the number of Holes in  $Z_p^j$  (i.e.  $h(Z_p^j)$ ).
9     if  $h(Z_p^j) = H$  then set  $MRS_i \leftarrow Z_p^j$ . Go to 26.
10    else find a port  $k$  for which  $D_i^j = \min\{Dist(s_i, Pt(Z_p^j)^k)\}$ ,  $\forall k \in$  Ports
11    if  $D_i^j < H - 1$  then
12      if  $D_i^j \geq h( LS(Pt(Z_p^j)^k) )$  then
13        Set  $MRS_i \leftarrow Z_p^j$ . Go to 26.
14      else go to 25.
15    end if
16    else if  $D_i^j = H - 1$  then
17      if  $h( LS(s_i) ) < H$  (for  $\forall LS(s_i)$ ) then
18        Set  $MRS_i \leftarrow Z_p^j$ . Go to 26.
19      else go to 25.
20    end if
21    else set  $MRS_i \leftarrow Z_p^j$ . Go to 26.
22  end if
23 end for
24 end for
25 Set  $MRS_i = \bigcup_{j=1}^{d(s_i)} Path(s_i, u_j)$ ;  $u_j \in LS(s_i)^j, Dist(s_i, u_j) \leq h( LS(s_i)^j )$ 
26 end for

```

Fig. 4. Pseudocode for calculating Maximum Reachability Spaces of robots.

TABLE IV
POSSIBLE DEADLOCK SITUATIONS CREATED BY TWO ROBOTS

Case	Description	Graphical Representation
1	$SP_i \subset SP_j$	
2	$SP_i \subset \neg SP_j \wedge s_i \neq g_j \wedge g_i \neq s_j$	
3	$SP_i \subset \neg SP_j \wedge s_i = g_j$	
4	$SP_i \subset \neg SP_j \wedge g_i = s_j$	
5	$SP_i = SP_j \wedge s_i = g_j \wedge g_i = s_j$	
6	$SP_i \not\subset SP_j \wedge SP_j \not\subset SP_i \wedge g_i \in SP_j \wedge g_j \in SP_i$	
7	$SP_i \not\subset SP_j \wedge SP_j \not\subset SP_i \wedge s_i \in SP_j \wedge s_j \in SP_i$	

D. An Example

The algorithm is run for the tree illustrated in Fig. 3. The shortest paths of robots, their relevant Zones and MRS are determined in Table V. Although it is verified that the goals of all robots lie within their MRS, for ensuring the solvability of the problem, deadlock situations must also be checked for all intersecting shortest paths of robots, as presented in Table VI. As a result, it becomes obvious that the problem has no solution because of deadlocks.

TABLE V
SHORTEST PATHS, \check{Z}_p , \check{Z}_s , AND MRS_i OF ROBOTS IN FIG. 3

Robot	Shortest Path	\check{Z}_p	\check{Z}_s	MRS
r_1	$\{v_1 \rightarrow j_1 \rightarrow v_2 \rightarrow v_3\}$	IZ_1	IZ_1	IZ_1
r_2	$\{j_1 \rightarrow v_6 \rightarrow j_2\}$	IZ_1, IIZ_1	IZ_1, IIZ_1	IZ_1
r_3	$\{v_3 \rightarrow v_2 \rightarrow j_1 \rightarrow v_1\}$	IZ_1	IZ_1	IZ_1
r_4	$\{j_2 \rightarrow v_6\}$	IZ_1, IIZ_1	IZ_1, IIZ_1	IIZ_1
r_5	$\{v_{10} \rightarrow j_2 \rightarrow j_3 \rightarrow v_{11}\}$	IIZ_1	IIZ_1	IIZ_1
r_6	$\{v_{14} \rightarrow j_2 \rightarrow j_3 \rightarrow v_{15} \rightarrow v_{16}\}$	IIZ_1	IIZ_1	IIZ_1
r_7	$\{j_3 \rightarrow j_2 \rightarrow v_6 \rightarrow j_1\}$	IIZ_1	IIZ_1	IIZ_1
r_8	$\{v_7 \rightarrow j_3 \rightarrow j_2 \rightarrow v_{14}\}$	IIZ_1	IIZ_1	IIZ_1
r_9	$\{v_{15} \rightarrow j_3 \rightarrow j_2 \rightarrow v_{10}\}$	IIZ_1	IIZ_1	IIZ_1
r_{10}	$\{v_{11} \rightarrow j_3 \rightarrow v_{15}\}$	IIZ_1	IIZ_1	IIZ_1
r_{11}	$\{v_{16} \rightarrow v_{15} \rightarrow j_3 \rightarrow v_7\}$	IIZ_1	IIZ_1	IIZ_1
r_{12}	$\{v_8 \rightarrow v_3 \rightarrow v_2\}$	\emptyset	\emptyset	$\{v_8, v_3, v_2, j_1\}$
r_{13}	$\{v_{12} \rightarrow v_{11} \rightarrow j_3\}$	IIZ_1	IIZ_1	$\{v_{17}, v_{12}, v_{11}, j_3\}$
r_{14}	$\{v_{17} \rightarrow v_{12}\}$	\emptyset	\emptyset	$\{v_{18}, v_{17}, v_{12}, v_{11}\}$
r_{15}	$\{v_{18} \rightarrow v_{13}\}$	IZ_4	IZ_4	$\{v_{13}, v_{18}, v_{17}, v_{12}\}$
r_{16}	$\{v_{13} \rightarrow v_{18} \rightarrow v_{17}\}$	\emptyset	IZ_4	$\{j_4, v_{13}, v_{18}, v_{17}\}$
r_{17}	$\{j_4 \rightarrow v_5 \rightarrow v_4\}$	IZ_4	IZ_4	IZ_4
r_{18}	$\{v_5 \rightarrow j_4 \rightarrow v_{13} \rightarrow v_{18}\}$	IZ_4	IZ_4	IZ_4
r_{19}	$\{v_4 \rightarrow v_5 \rightarrow j_4 \rightarrow v_9\}$	IZ_4	IZ_4	IZ_4

TABLE VI
DEADLOCK SITUATIONS OF ROBOTS IN FIG. 3

Robot	Compared With	Deadlock Situation	Case
r_1	r_{12}	$g_1 \in SP_{12} \wedge g_{12} \in SP_1$	6
r_2	r_4	$SP_4 \subset \neg SP_2 \wedge s_4 = g_2$	3
	r_7	$SP_2 \subset \neg SP_7 \wedge s_2 = g_7$	3
r_5	r_{13}	$g_5 \in SP_{13} \wedge g_{13} \in SP_5$	6
r_{15}	r_{18}	$SP_{15} \subset \neg SP_{18} \wedge s_{15} = g_{18}$	3
	r_{16}	$SP_{15} \subset \neg SP_{16} \wedge g_{15} = s_{16}$	4

VI. DISCUSSION

As explained above, the algorithm for verifying whether a given problem on a tree has a solution is performed in three phases: (1) calculating Influence Zones, (2) determining the Maximum Reachability Space of each robot, and (3) checking possible Deadlocks in robots' motions. The time complexity of the algorithm is in $O(n)$ in phase 1, and $O(m)$ in phases 2 and 3, where n and m are the number of tree's vertices and robots, respectively.

For experimentation, we designed various rooted trees (RT) with different branching factors (b) and depths (d), and ran the algorithm on a PC with 2.8 GHz Intel CPU and 1 GB of RAM. Each problem was solved 10 times for random start and goal configurations. The average runtimes are:

Problem in Fig. 3: $n = 22$, $m = 19$, time = 0.078 sec.
 RT: $b = 2$, $d = 5$, $n = 63$, $m = 50$, time = 0.748 sec.
 RT: $b = 4$, $d = 3$, $n = 85$, $m = 70$, time = 1.327 sec.
 RT: $b = 3$, $d = 4$, $n = 121$, $m = 100$, time = 2.861 sec.
 RT: $b = 5$, $d = 3$, $n = 156$, $m = 150$, time = 5.191 sec.

Since we didn't find any previous work in the literature with assumptions as general as ours, we had no benchmarks to make comparisons, and therefore evaluated the efficiency of our findings in terms of time and memory by comparing our results with the number of operations required for thoroughly searching the configuration space. Investigating the solvability of an MRMP problem of m robots on a tree with n vertices

through exhaustive enumeration will require $n!/(n-m)!$ different permutations of robots to be checked, which is far beyond the time order of the presented algorithm. Also, verifying whether a tree is SG^m would require $\sum_{i=1}^m (n!/(n-i)!)^2$ operations to be checked, for any initial and final configurations, which is again exponentially time consuming.

VII. CONCLUSION

For graph-based multi robot routing, it is necessary to ensure that the graph has a proper topology and sufficient number of vertices (relative to the number of robots) to enable planning of robots' moves. In this paper a number of novel contributions are proposed in the field of MRMP on trees: the concepts Solvable, Minimal Solvable, and Partially Solvable Trees are introduced. Considering that the complexity of graph searching operations is directly influenced from the graph size, finding Minimal Solvable Trees would significantly ease the motion planning task for multiple robots on trees. Also, a new algorithm with linear time complexity is proposed for deciding whether a multi robot motion planning problem has a solution on a tree-type graph, without explicitly solving it. The algorithm also computes the Maximum Reachability Space of the robots which can be used to determine the whole class of solvable problems on the tree. The result of this algorithm is valuable especially for designing solvable trees or evaluating an existing network or workspace.

In a complementary work, we have generalized the concepts of ST^m , PST^m , and MST^m to Solvable Graphs (SG^m), Partially Solvable Graphs (PSG^m), and Minimal Solvable Graphs (MSG^m), respectively, for cyclic graphs which contain at least one cycle, and have proposed similar algorithms for determining their solvability [11].

REFERENCES

- [1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Boston, 2005.
- [2] C. M. Clark, "Probabilistic Road Map Sampling Strategies for Multi-Robots Motion Planning," *Rob. Auton. Sys.*, Vol. 53, pp. 224-265, 2005.
- [3] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proc. IEEE Int. Conf. Rob. Autom.*, 1989, pp. 484-489.
- [4] J. van den Berg and M. Overmars, "Prioritized motion planning for multiple robots," in *Proc. IEEE/RSJ IROS 2005*, pp. 2217-2222.
- [5] R. M. Wilson, "Graph puzzles, homotopy, and the alternating group", *Journal of combinatorial Theory*, Series B, Vol. 16, pp. 86-94, 1974.
- [6] D. Kornhauser, G. Miller, and P. Spirakis, "Coordinating pebble motion on graphs, the diameter of permutations groups and applications", in *Proc. 25th IEEE Symp. Found. of Comp. Science*, (1984), pp. 241-250.
- [7] V. Auletta, A. Monti, D. Parente and G. Persiano, "A Linear-Time Algorithm for the Feasibility of Pebble Motion on Trees", *Algorithmica*, Vol. 23, No. 3, pp. 223-245, 1999.
- [8] M. R. K. Ryan, "Exploiting subgraph structure in multi-robot path planning", *J. Artificial Intell. Research*, Vol. 31, pp. 497-542, 2008.
- [9] R. Diestel, *Graph Theory*, Springer-Verlag, New York, 2000.
- [10] V. Auletta, D. Parente and G. Persiano, "A New Approach to Optimal Planning of Robot Motion on a Tree with Obstacle", in *Proc. 4th European Symposium on Algorithms (ESA)*, Spain, 1996, pp. 25-27.
- [11] E. Masehian, H. Samadian, and F. Daneshzand, "Analysis of motion feasibility of multiple mobile agents on graphs," in *Proc. 4th International Conference on Information Technology (ICIT) 2009*, Jordan.