Proceeding of the 2nd
RSI/ISM International Conference on Robotics and Mechatronics
October 15-17, 2014, Tehran, Iran

Computing a configuration skeleton for motion planning of two round robots on a metric graph

Dr Vitaliy Kurlin

Department of Mathematical Sciences
Durham University, Durham DH1 3LE, UK
Email: vitaliy.kurlin@gmail.com

Dr Marjan Safi-Samghabadi

Tehran, Iran
Email: marjan.safi.samghabadi@gmail.com

***Abstract*—A connected metric graph G with n vertices and without loops and multiple edges is given as an $n \times n$-matrix whose entry a_{ij} is the length of a single edge between vertices $i \neq j$. A robot in the metric graph G is the metric ball with a center $x \in G$ and a radius $r > 0$.**

The configuration space $OC(G,r)$ of 2 ordered robots in G is the set of all centers $(x,y) \in G \times G$ such that x,y are at least $2r$ away from each other. We introduce the configuration skeleton $CS(G,r) \subset OC(G,r)$ that captures all connectivity information of the larger space $OC(G,r)$.

We design an algorithm of time complexity $O(n^2)$ to find all connected components of $OC(G,r)$ that are maximal subsets of all safe positions (x,y) connectable by collision-free motions of the two round robots.

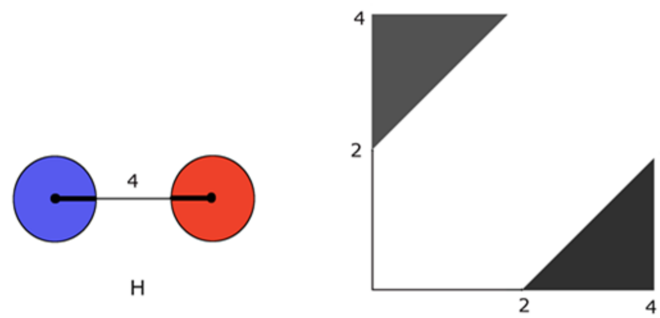

Fig. 1. Left: 2 robots of radius 1 in $H = [0,4]$. Right: the space $OC(H,1)$.

I. Introduction

A. Round robots on a metric graph

Examples of metric graphs in practice are magnetic tapes on a floor, train tracks, blood vessels or trajectories of particles. Vertices of such a graph are junctions where more than two branches meet.

Our modeled robot could be an automatic train that moves on the railway. Each time the train moves through a junction, all other nearby trains should be kept away from the junction. On a small scale, a robot can be a string shaped device that moves through a blood vessel and it has the ability to spread out at a junction v to arcs of length at most r within all branches attached to the junction v.

More formally, a non-oriented graph G consists of a finite set of vertices and a finite set of edges, where each edge e connects two vertices, has a fixed length l and is isometric to the segment $[0,l]$ in the real line. Then G becomes a metric space with the distance $d(x,y)$ equal to the length of a shortest path between $x,y \in G$.

During a motion of two robots of a radius r on G the distance between their centers x,y should remain at least $2r$. Then such a configuration $(x,y) \in G \times G$ is called *safe* or *collision-free*.

For a graph G in the plane, it is convenient to draw a robot as a round disk, see Fig. 1. However we consider abstract graphs not embedded into any space and the following more abstract model of a robot that is always within G. Namely, the metric ball with a center $x \in G$ and a radius $r > 0$ is the set $\{y \in G : d(x,y) \leq r\} \subset G$ of those points in the graph G that are within the distance r from the center x.

B. The configuration space of two round robots

The **o**rdered **c**onfiguration space $OC(G,r)$ of two robots of the same radius r in a metric graph G consists of all pairs $(x,y) \in G \times G$, where x,y are the centers of the robots with $d(x,y) \geq 2r$.

The **u**nordered **c**onfiguration space $UC(G,r)$ of two unlabeled robots in G, is the quotient space $OC(G,r)/S_2$, where S_2 is the symmetric group acting on $OC(G,r)$ by permuting 2 robots.

We consider collision-free motions, but we allow the robots to touch each other when $d(x,y) = 2r$. Therefore, the configuration space $OC(G,r)$ is compact. We exclude the case when $r = 0$ because the inequality $d(x,y) \geq 0$ allows collisions.

In Fig. 1, we consider two robots on the straight segment $H = [0,4]$ in the real line. The configuration space $OC(H,1)$ consists of two symmetric triangles in the right hand side picture of Fig. 1.

C. Our contribution: the configuration skeleton CS

We call a space Z (path-)*connected* if we can join any two points $x,y \in Z$ by a continuous path $\alpha : [0,1] \subset Z$, where

$x = \alpha(0)$, $y = \alpha(1)$. The properties of connectedness and path-connectedness are equivalent for simplicial complexes including graphs. The configuration space $OC(H,1)$ in Fig. 1 consists of two connected components.

We introduce a 1-dimensional subgraph of the 2-dimensional space $OC(G,r)$ that is called the **c**onfiguration **s**keleton $CS(G,r)$. We prove in section III that the inclusion $CS(G,r) \subset OC(G,r)$ establishes a 1-1 map (bijection) between all connected components.

Hence, computing connected components of the space $CS(G,r)$ reduces to the simpler connectivity problem of the smaller skeleton $CS(G,r)$. The set of vertices in $CS(G,r)$ is the same as the set of vertices in $OC(G,r)$. So the algorithm can output all pairs $(u,v) \in OC(G,r)$ of vertices u,v that are in the same connected component.

Practically, if two pairs $(x,y),(w,z) \in OC(G,r)$ are in different components, then it is not possible to swap these robots without collisions. Having computed the skeleton $CS(G,r)$, we can decide if two given configurations $(x,y),(w,z)$ can be connected by a collision-free motion within the original configuration space $OC(G,r)$.

D. Previous work on configuration spaces

The tradition of robots following a guide-path of magnetic tapes on the floor has led to modeling the problem of studying the motion of robots on graphs. The simplest model when robots are zero-sized points has been studied in considerably short period of time. Topological invariants of configuration spaces of robots on graphs including the Euler characteristic, the fundamental group, homology and cohomology groups are the main motivation of most research carried out in this topic.

A. Abrams and R. Ghrist in [9] considered configuration spaces of the Automated Guided Vehicles (AGVs) in a warehouse. K. Barnett and M. Farber in [5] studied homology and cohomology of the configuration space $OC_n(G,0)$ of n zero-sized robots on a graph G. Consecutively, M. Farber and E. Hanbury in [6] investigated the homology groups of $OC_n(G,0)$, where G is a generalized mature graph.

D. Farley and L. Sabalka have studied the cohomology of n zero-sized robots on a tree in [7]. A. Abrams, D. Gay, and V. Hower show that the discretized configuration space of n points in the $k-$simplex is homotopy equivalent to a wedge of spheres of dimension $kn+1$ in [1]. K. H. Ko and H. W. Park in [10] computed the first homology of the configuration space $UC_n(G,0)$ of n zero-sized robots on G.

V. Kurlin in [11] designed an algorithm to write down a presentation with explicit generators and relations for the fundamental group of $OC_n(G,0)$. The key idea is to update a presentation according to a Seifert - van Kampen construction when we start from a simple graph G and then add edges to G one by one. Every generator is realized as a collision-free motion of n zero-sized robots in the graph G.

One of the applications of configuration spaces of robots on graphs in physics is explored in [4] and [3]. In [4], JM Harrison, JP Keating, JM Robbins consider 2 unordered spinless particles on a quantum graph G. They introduce a new way to study quantum statistics. This result is applied to topological quantum computing, topological insulators, the fractional quantum Hall effect, superconductivity and molecular physics. In [3], Jonathan M. Harrison, Jonathan P. Keating, Jonathan M. Robbins, Adam Sawicki study the relation between quantum statistics and the connectivity of quantum graph G. They have also computed topological gauge potentials for 2-connected quantum graphs.

K. Deeley was probably the first who considered in [2] round robots of a positive radius a connected graph G. He proved that for any small enough radius $r > 0$ the configuration space $OC_n(G,r)$ of n round robots is homotopy equivalent to the space $OC_n(G,0)$ of zero-sized robots. We consider the same model of a robot as a metric ball and answer the harder question whether two configurations in $OC(G,r)$ can be connected by a collision-free motion for any radius $r > 0$.

II. Connectivity of configuration spaces

A configuration space $OC(G,r)$ consists of all pairs $(x_1,x_2) \in G^2$ where x_1, x_2 are the centers of robots such that the robots do not collide. When we say $OC(G,r)$ is connected, we mean there is a path in $OC(G,r)$ between any two configurations of $OC(G,r)$. Considering a graph with at least one vertex of degree greater than 2, we can show that $OC(G,r)$ is connected if the length of all edges is not less than $2r$. See Fig. 2.

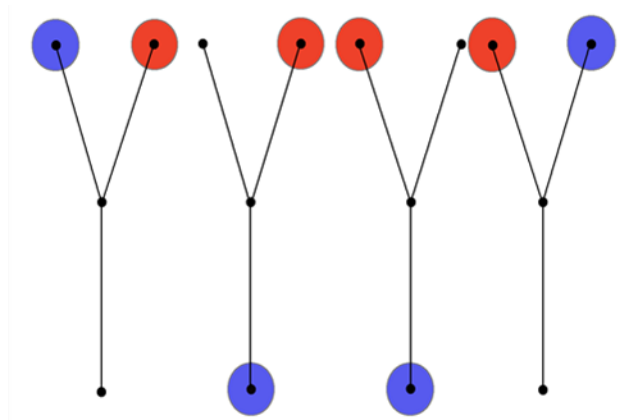

Fig. 2. Robots swap without collisions on the graph.

Example 1. *Consider the connected graph G shown in Fig. 3.*

Case. 1 *If* $0 < r \leq 1$, *the configuration space* $OC(G,r)$ *is connected and shown in Fig. 3 [right]. We can find a continuous motion between any two configurations of* $OC(G,r)$. *The blue area shows the configurations when the robots are on different edges. The yellow area shows the configurations when the robots are on the same edge. Finally,* *the black skeleton shown in Fig. 3[right-bottom] is for the configurations that both robots are on vertices or one robot is fixed on a vertex and the second robot moves along one edge. The black skeleton represents configuration skeleton $CS(G,r)$.*

Case. 2 *If $1 < r \leq 2$, the configuration space $OC(G,r)$ is not connected because there is not any continuous motion from (v_1,v_2) to (v_1,v_3). Since fixing a robot on a vertex of degree one stops the second robot to move to the vertex v, the robots only perturb on the vertices of degree one. Therefore, the configuration space consists of 6 connected components as shown in Fig. 3[bottom-left]. The number of connected components in this case is equal to the number of vertices in $OC(G,r)$. In other words by fixing a robot at a vertex, the second robot cannot move along one edge. So the configuration skeleton is limited to 6 vertices where both robots are fixed at vertices in G.*
For $2 < r$, the configuration space will be empty, therefore $CS(G,r)$ is also empty.

radius	*No. of connected components of CS*
$0 < r \leq 1$	*1*
$1 < r \leq 2$	*6*
$2 < r$	*0*

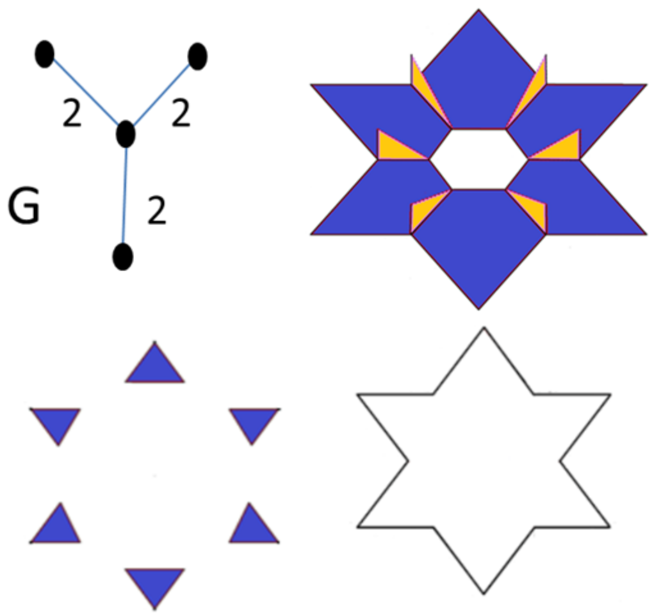

Fig. 3. (The graph G, (left) the configuration space $OC(G,r)$ for $r \leq 1$ (right). The configuration skeleton $CS(G,r)$ for $r \leq 1$ (right-bottom). $OC(G,r)$ for $1 < r \leq 2$ (left-bottom).

Example 2. *Consider the connected graph Q and two robots as shown in Fig. 4.*

Case 1. *If $0 < r \leq 1$, the configuration space $OC(Q,r)$ is connected and shown in Fig. 4 [right]. The blue cylindrical area shows the configurations when a robot is on the cycle and the second robot is on the hanging edge. The yellow cylinder shows the configurations when both robots are on the cycle. The red triangles are when both robots are on the hanging edge. The black skeleton $CS(Q,r)$ is for the configurations that both robots are on vertices or one robot is fixed on a vertex and the second robot moves along one edge.*

Case 2. *If $1 < r \leq 2$, the configuration space consists of 2 disjoint vertices (v_1,v_3), (v_3,v_1). The robots collide when both are on the cycle, or when both are on the same edge. In fact, the number of connected components of $OC(Q,r)$ is maximum 2 by changing the robots radii.*

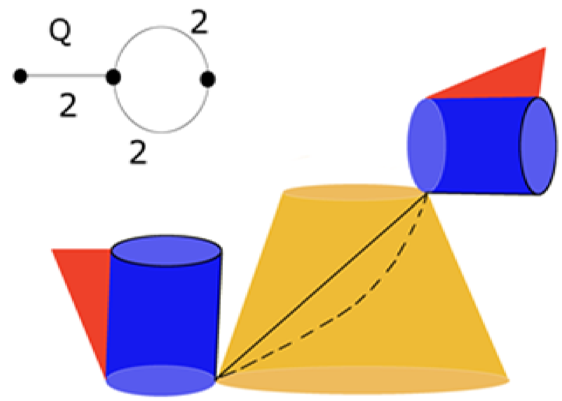

Fig. 4. (left) The graph Q, (right) the configuration space $OC(Q,r)$ for $0 < r \leq 1$.

As we have seen in the examples above, constructing the configuration space $OC(G,r)$ for any connected graph G is not easy. In the following section, we will see a technique to find the number of connected components of $OC(G,r)$ without constructing the configuration space.

III. CONFIGURATION SKELETON

In this section, we define the configuration skeleton $CS(G,r)$ of $OC(G,r)$. We will show that the number of connected components of the following configuration skeleton is equal to the number of connected components of $OC(G,r)$. Since the set of vertices in $CS(G,r)$ is equal to the set of vertices in $OC(G,r)$, by constructing $CS(G,r)$, we show the vertices that are in the same connected component. A cycle C in G is a sequence of distinct vertices $v_1, \dots, v_k$ such that any v_i, v_{i+1} (in the cyclic order) are adjacent. The following definition introduces the concept of the configuration skeleton $CS(G,r)$.

Definition 3. *Let G be a connected metric graph and a cycle C in G is a sequence of distinct vertices $v_1, \dots, v_k$ such that any v_i, v_{i+1} (in the cyclic order) are adjacent. We assume that any vertex on a cycle $C \subset G$ has a diametrically opposite vertex, otherwise, we add the diametrically opposite vertex of degree two to the cycle C. The* configuration skeleton *of $OC(G,r)$, denoted as $CS(G,r)$, is the following combinatorial graph whose vertices are all pairs (u,v), where u, v are vertices of G and the distance $d(u,v) \geq 2r$.*
(1) *We connect vertices (v,u), (w,u) by an edge in $CS(G,r)$ if*

v, w are connected by an edge in G. (Similarly, we connect the vertices (u,v), (u,w).)
(2) We connect vertices (u,v), (w,z) by an edge in $CS(G,r)$ if,

- *u, w, are adjacent vertices on a cycle $C \subset G$, and*
- *v, z, are adjacent vertices on the same cycle $C \subset G$, and*
- *$d(u,z) < 2r$, $d(v,w) < 2r$.*

The following example illustrates Definition 3.

Example 4. *Consider graph G as shown in Fig. 5 and the robots $(x, \frac{3}{2})$, $(y, \frac{3}{2})$. The graph G consists of one cycle of length 6. To create the sub graph $CS(G, \frac{3}{2})$, first we add the diametrically opposite vertices d, c to the vertices a, b, respectively as shown in Fig. 5 [right]. This is to ensure we have vertices of the furthest distance on the cycle.*

When both robots are at the vertices in G and they do not collide, we get the vertices of $OC(G, \frac{3}{2})$. For example, when $(x, \frac{3}{2})$ is at u and $(y, \frac{3}{2})$ is at v, we get the vertex (u,v) in $OC(G, \frac{3}{2})$ which is shown as a vertex in CS in Fig. 5 [bottom]. But having $(x, \frac{3}{2})$ at u and $(y, \frac{3}{2})$ at a in G is not collision-free, so (u,a) is not a vertex in G. Considering the condition in Definition 3, the set of vertices in the sub graph $CS(G, \frac{3}{2})$ is equal to the set of vertices in $OC(G, \frac{3}{2})$.

After finding the vertices in $CS(G, \frac{3}{2})$, we connect them by using Definition 3(1),(2). For instance, we connect the vertices (u,v), $(u,w) \in CS(G, \frac{3}{2})$ since v, w are adjacent in G. Similarly, we connect (v,u), (w,u) using Definition 3(1). All blue edges in the graph CS in Fig. 5 are constructed in this way.

Considering a robot on a and the other one on d, we get (a,d) in the configuration skeleton $CS(G, \frac{3}{2})$. Informally, in this example the robots can be at any points on the cycle as far as the points are diametrically opposite. Since d is too close to the adjacent vertices b, c (similarly, a is too close to the adjacent vertices b, c on the cycle), Definition 3(1) does not cover this case. we connect (a,d), (b,c) in $CS(G, \frac{3}{2})$ by Definition 3(2) as a is adjacent to b and d is adjacent to c.

in the following section we will explain how the above configuration skeleton defines a particular motion in $OC(G, \frac{3}{2})$.

Theorem 5. *For any connected graph G and a radius $r > 0$, the inclusion $CS(G,r) \subset OC(G,r)$ induces a 1-1 correspondence between connected components of the configuration space $OC(G,r)$ and its smaller skeleton $CS(G,r)$. Hence, for vertices u,v,w,z of G, the configurations $(u,v),(w,z)$ can be connected by a collision-free motion in $OC(G,r)$ if and only if they can be connected by a path in the configuration skeleton $CS(G,r)$.*

IV. Elementary motions of robots

Before defining what an elementary motion is, we explain the importance of vertices in $CS(G,r)$. Let robots be at any points on the graph G without collision, we have three cases: (1) both robots be at vertices in G, (2) both robots be on some edges in G and (3) a robot be at a vertex and the other robot be on an edge. We claim we can achieve case (1) from

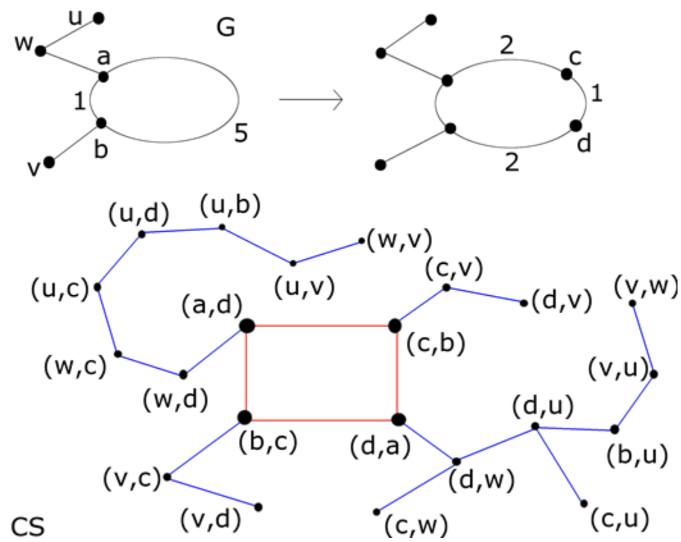

Fig. 5. First we add diametrically opposite vertices to all vertices on the cycle (right). The configuration skeleton has one connected component (bottom).

the cases (2), (3) by pushing robots on an edge to a possible close vertex. Considering the robots are not on a cycle in G, we can push away the robots until they reach a vertex. When the robots are on a cycle, by pushing the robots away on the cycle, they reach a vertex since we have a diametrically opposite vertex to any vertex. This is important since it shows there is a path from any configuration to a vertex in $OC(G,r)$.

Definition 6. *An* elementary motion *in $OC(G,r)$ is defined as follows.*
(1) Let u, v, $w \in G$ be vertices. For example see Fig. 5. If

- *v, w are connected with an edge in G, and*
- *$d(v,w) \geq 2r$,*

then the motion from (u,v) to (u,w) is called elementary. *This means the first robot is fixed at vertex u and the second robot moves from v to the adjacent vertex w. (Similarly, we define an* elementary motion *from (v,u), (w,u)).*
(2) Let the vertices a, b, c, d be on a cycle of G. For instance, see Fig. 5. If

- *$d(a,d) \geq 2r$, $d(b,c) \geq 2r$*
- *a is adjacent to b and d is adjacent to c*

then we can move the first robot from a to b and simultaneously, the second robot from d to c in the same direction on the cycle without collisions. This motion is also called an elementary motion.

As we discussed in the beginning, the set of vertices in $CS(G,r)$ is the same as the set of vertices in $OC(G,r)$. By definition above every edge in $CS(G,r)$ between two vertices is an elementary motion between the same configurations in $OC(G,r)$. Any two edges in $CS(G,r)$ may only meet at vertices.

The following Theorem states the main result.

The algorithm given in the following section computes the number of connected components of $CS(G,r)$ and by Theorem 5, such number is equal to the number of connected components of $OC(G,r)$.

V. ALGORITHM

The algorithm **input** is the $n \times n$ matrix of a graph G with n vertices and each non-zero entry a_{ij} is the length of the edge between two adjacent vertices i, j. If the vertices i, j are not adjacent, we have $a_{ij} = 0$. The **output** is the graph of the configuration skeleton $CS(G,r)$ of 2 robots on G. Such a graph consists of all vertices in $OC(G,r)$. The number of connected components of $CS(G,r)$ is equal to the number of connected components of $OC(G,r)$, so the output graph illustrates the vertices in $OC(G,r)$ that are in the same connected component.

Step 1. The algorithm lists all the cycles of G with time complexity $O(n^2)$, where n is the number of vertices of G. See [8]. By computing the distance between vertices on such cycles, it adds diametrically opposite vertices to each vertex on a cycle. Therefore, the algorithm creates a new matrix for the graph G by adding new vertices and adjusting the length of corresponding edges.

Step 2. Consider all pairs (i,j), where i, j are vertices in G. If G consists of n vertices, we have $n(n-1)$ pairs. The algorithm constructs a $(n^2-n) \times (n^2-n)$ adjacency matrix $CS(G,r)$, where the rows and the columns are the pairs (i,j). Since we can arrange the rows and columns arbitrarily, we arrange rows and columns in an ascending order: $(1,1)$, $(1,2)$, $(1,3),\dots,(2,1)$ $(2,2)$, $(2,3),\dots,(n^2-n,n^2-n)$. We insert copies of the adjacency matrix of the graph G in $CS(G,r)$ in the following way:

- we insert the first copy of $n \times n$ adjacent matrix A of the graph G in the first n rows and columns. We insert the second copy of A in $(n+1)$ to $2n$ rows and columns. We insert the third copy of A in $(2n+1)$ to $3n$ rows and columns, etc.
- the remaining entries of $CS(G,r)$ are $n \times n$ diagonal matrices, where the rows are from $(i,1)$ to (i,n) and the columns are from $(j,1)$ to (j,n) and the entry of the diagonal is 0 if i, j are not adjacent in A and is 1 if i, j are adjacent in A.

This step is done with time complexity $O(n)$.

Step 3. The algorithm checks the value of each entry in the matrix of the given graph to be less than $2r$. We remove the rows and the columns for vertices (i,j) in $CS(G,r)$ that $d(i,j) < 2r$ to avoid collisions. This step is done with time complexity $O(n^2)$.

Step 4. The algorithm lists all connected vertices in $CS(G,r)$ in some separate component sets. Then starts from one set, finds the vertex (i,j) such that i, j are on the same cycle. Then checks if the adjacent vertex k to i and the adjacent vertex l to j are on the same cycle and checks $d(k,i) < 2r$, $d(l,j) < 2r$. Finally, looks for (k,l) in other component sets. If all previous steps are successful, in matrix $CS(G,r)$, changes the entry of the row (i,j) and the column (k,l) to 1 and does the same process in the next component set. In other words, we only need to find one connection between any two component sets. If the previous steps are not successful, the algorithm will check for the next element of the first component set until reaching the success, otherwise, the algorithm introduces that component set disconnected from other component sets.

Step 5. The algorithm plots $CS(G,r)$ or introduces the final component sets.

The configuration skeleton $CS(G,r)$ consists of the same number of connected components as the configuration space $OC(G,r)$. The vertices in $CS(G,r)$ are the same as the vertices in $OC(G,r)$, so $CS(G,r)$ shows which vertices of $OC(G,r)$ are in the same connected component.

Lemma 7. *The time complexity of the construction of the configuration skeleton $CS(G,r)$ is at most $O(n^2)$.*

Proof: The algorithm constructs $CS(G,r)$ in 5 steps. The time complexity of each step is at most $O(n^2)$, so the sum of all the steps time complexity is $O(n^2)$. ■

Lemma 8. *The time complexity of the construction of the configuration skeleton $CS(T,r)$ is at least $O(n^2)$.*

Proof: The configuration skeleton $CS(T,r)$ is the simplest case since a tree T is connected and does not consist of any cycles. The algorithm constructs $CS(T,r)$ in 3 steps: step 2, step 3 and step 5. Since step 3 is done with time complexity $O(n^2)$ then the time complexity of the algorithm is $O(n^2)$. ■

As shown in Example. 1 [table], for different ranges of radii we get different number of connected components for $CS(G,r)$. Such algorithm states the number of connected components of $CS(G,r)$ for all ranges of robot radii.

VI. ALGORITHM JUSTIFICATION

In the following Lemma we explain the possibility of replacing any collision-free motions between two configurations in $OC(G,r)$ by a finite sequence of elementary motions. Lemma 9 plays the key role in proving Theorem 5.

Lemma 9. *Any collision-free motion $(x(t),y(t))$, $0 \le t \le 1$, where $x(0)$, $y(0)$, $x(1)$, $y(1)$ are vertices of G, can be replaced by a finite sequence of elementary motions.*

Proof: We prove the theorem by induction on the number k of vertices in G that at least one of the robots visits during the motion $(x(t),y(t))$, $0 \le t \le 1$. Vertices are counted with multiplicities, i.e. when in a motion a robot visits the same vertex m times over $0 < t < 1$, then we count this vertex m times. But the initial and the final vertices $x(0)$, $y(0)$, $x(1)$, $y(1)$ are not counted.

Induction base: $(k = 0)$ If the robots do not visit any vertices over $0 < t < 1$, then robot 1 moves along one edge and robot 2 moves simultaneously along another edge. There are the following two cases.

Case (1) : let $d(x(0),y(1)) \geq r_1 + r_2$ or $d(x(1),y(0)) \geq r_1 + r_2$. Then the motion from $(x(0),y(0))$ to $(x(1),y(1))$ can be replaced by two elementary motions by Lemma 4.11 [12], where $u = x(0)$, $v = y(0)$, $w = x(1)$, $z = y(1)$.

Case (2) : let $d(x(0),y(1)) < r_1 + r_2$ and $d(x(1),y(0)) < r_1 + r_2$. Then by Definition 6, the motion from $(x(0),y(0))$ to $(x(1),y(1))$ is elementary. So the induction base $k = 0$ is complete.

Inductive assumption: let the theorem hold for all motions when both robots visit at most k vertices of G, counted with multiplicities.

Inductive step: We prove the theorem for a motion when both robots visit exactly $k+1$ vertices of G. We consider the time interval from 0 to the first moment $t \in (0,1)$, when one of the robots reaches a vertex, say robot 1. So robot 1 moves from the vertex $x(0)$ to an adjacent vertex $x(t)$, and robot 2 moves from the vertex $y(0)$ to a point $y(t)$, not a vertex. There are no vertices between $y(0)$, $y(t)$. We have the following cases.

Case(1) : let $d(x(t),y(0)) \geq r_1 + r_2$.

- We fix robot 2 at $y(0)$ and move robot 1 from $x(0)$ to $x(t)$. This elementary motion from $(x(0),y(0))$ to $(x(t),y(0))$ is collision-free since $y(0)$ is far away from both points $x(0)$ and $x(t)$.
- Then we fix robot 1 at $x(t)$ and move robot 2 from $y(0)$ to $y(t.)$ This motion from $(x(t),y(0))$ to $(x(t),y(t))$ is collision-free since $x(t)$ is far away from both points $y(0)$ and $y(t)$.

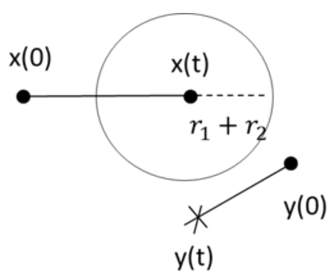

Fig. 6. The figure illustrates case (1), when $y(0)$, $x(t)$ are far away.

After that the robots move from $(x(t),y(t))$ to $(x(1),y(1))$ as in the original motion. During the motion from $(x(t),y(0))$ to $(x(1),y(1))$, the robots visit only k vertices because the vertex $x(t)$ is not counted anymore as the initial position of robot 1. So the inductive step is finished in case (1).

We apply Lemma 4.12 [12] for $v = y(0)$, $w = x(t)$, $z = y(t)$. The assumptions of Lemma 4.12 [12] hold since

- $x(0)$, $y(0)$, $x(t)$ are vertices of G, and
- the point $y(t) \in G$ is not a vertex, and
  the vertex $x(0)$ is adjacent to $x(t)$, and
- there is no vertex between $y(0)$ and $y(t)$, and
- we have $d(x(t),y(0)) < r_1 + r_2$.

Let q be the adjacent vertex to $y(0)$ by the edge that contains $y(t)$. The condition

$$d(w,z) = d(x(t),y(t)) \geq r_1 + r_2$$

in Lemma 4.12 [12] holds, because the robots at time t do not collide. Lemma 4.12 [12] implies that $d(q,x(t)) \geq r_1 + r_2$, for $w = x(t)$, $v = y(0)$, $z = y(t)$.

Case(2) : let $d(x(t),y(0)) < r_1 + r_2$ and $d(q,x(0)) \geq r_1 + r_2$.

- We fix robot 1 at $x(0)$ and push robot 2 from $y(0)$ to q. So the elementary motion $(x(0),y(0))$ to $(x(0),q)$ is collision-free since $x(0)$ is far away from both points $y(0)$, q.
- Then we fix robot 2 at q and push robot 1 from $x(0)$ to $x(t)$. The elementary motion from $(x(0),q)$ to $(x(t),q)$ is collision-free since q is far away from $x(0)$, $x(t)$.
- We now fix robot 1 at $x(t)$ and push robot 2 back from q to $y(t)$. This motion from $(x(t),q)$ to $(x(t),y(t))$ is collision-free since $x(t)$ is far away from both q, $y(t)$.

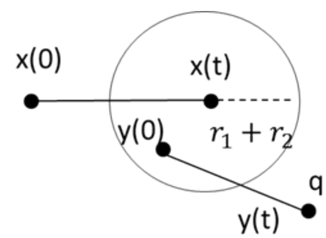

Fig. 7. Case (2), when $y(0)$, $x(t)$ are close, but q, $x(0)$ are far away.

After that we have the original motion from $(x(t),y(t))$ to $(x(1),y(1))$. During the motion from $(x(t),q)$ to $(x(1),y(1))$, the robots visit only k vertices because the vertex $x(t)$ is not counted anymore as the initial position of robot 1. So the inductive step is finished in case (2).

Case (3) : Let $d(x(t),y(0)) < r_1 + r_2$ and $d(q,x(0)) < r_1 + r_2$. Then $x(0)$, $y(0)$, $x(t)$, $y(t)$ are on a topological circle $C \subset G$.

- Then we move robot 1 from $x(0)$ to $x(t)$, simultaneously, we move robot 2 from $y(0)$ to q. By Definition 6, the elementary motion from $(x(0),y(0))$ to $(x(t),q)$ is collision-free since $x(t)$, q are far away.
- Then we fix robot 1 at $x(t)$ and move robot 2 back from q to $y(t)$. The motion from $(x(t),q)$ to $x(t),y(t)$ is collision-free since $x(t)$, $y(t)$ are far away.

After that we have the original motion from $(x(t),y(t))$ to $(x(1),y(1))$. During the motion from $(x(t),q)$ to $(x(1),y(1))$, the robots visit only k vertices because the vertex $x(t)$ is not counted anymore as the initial position of robot 1. So the inductive step is finished in case (3).

■

Proof of Theorem 5. Since the set of vertices in the configuration space $OC(G,r)$ is equal to the set of vertices in

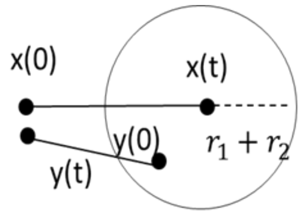

Fig. 8. Case (3), when $y(0)$, $x(t)$ are close, and q, $x(0)$ are close too.

the configuration skeleton $CS(G,r)$, and any motion between vertices in $OC(G,r)$ can be replaced by a finite sequence of elementary motions by Lemma 9. Therefore, if we have two vertices in the same connected component in $OC(G,r)$, then those vertices are in a connected component in $CS(G,r)$. Reversely, if there is an edge between two vertices in $CS(G,r)$, the corresponding configurations are connected in $OC(G,r)$ by an elementary motion. ■

VII. Conclusion and Future Work

In previous sections we defined the configuration space $OC(G,r)$ of 2 robots with radii r_1, r_2 on a connected graph G where each edge has a fixed length. Our main interest is to compute the number of connected components in $OC(G,r)$. In few examples, we have shown that constructing $OC(G,r)$ is not easy. Therefore, we defined a configuration skeleton $CS(G,r)$ of $OC(G,r)$ with the same number of connected components and the Algorithm V computes the number of connected components of $CS(G,r)$. Theorem 5 is a formal statement to prove such claim.

Briefly speaking, in this paper we introduced a new technique to compute the number of connected components by the algorithm with $O(n^2)$ time. In this technique we highlight the configurations when at least one robot is at a vertex in G. Since the number of such configurations is finite we can implement it in a fast algorithm. Then we show that all other configurations in $OC(G,r)$ can continuously move to such highlighted configuration.

In this topic, there are several interesting invariants to discover. For example, the fundamental group of $CS(G,r)$ for particular graphs such as K_5, $K_{3,3}$ is not isomorphic to the fundamental group of $OC(G,r)$. Therefore, a formal statement to describe the fundamental group of $OC(G,r)$ remains open.

We can also progress this topic by considering finite number of robots on a connected graph G and investigate the behavior of $OC_n(G,r)$ by finding different invariants of such configuration space.

Another interesting exploration can be done by considering robots on a obstacle-free plane (or even a plane including several obstacles). Then define the configuration space of such case that provides a higher freedom for the motion of AGVs on the real factory floor.

References

[1] A. Abrams, D. Gay, and V. Hower, *Discretized configurations and partial partitions*. Proc. Amer. Math. Soc. 141 (2013), 1093-1104.

[2] K. Deeley, *Configuration spaces of thick particles on a metric graph*. Algebraic & Geometric Topology 11 (2011) 1861-1892.

[3] J. M. Harrison, J. P. Keating, J. M. Robbins, A. Sawicki, *n-particle quantum statistics on graphs*. Communications in Mathematical Physics 330 (2014), 1293-1326.

[4] JM Harrison, JP Keating, JM Robbins, *Quantum statistics on graphs*. Proc. R. Soc. A 467(2125), 21223 (2011).

[5] K. Barnett and M. Farber, *Topology of configuration space of two particles on a graph, I*. Algebraic & Geometric Topology 9 (2009) 593-624.

[6] E. Hanbury and M. Farber, *Topology of configuration space of two particles on a graph, II*. Algebraic & Geometric Topology 10 (2010) 2203-2227.

[7] D. Farley and L. Sabalka, *On the cohomoloy rings of tree braid groups*. Journal of Pure and Applied Algebra 212 (2008), 53-71.

[8] R. Ferreira, R. Grossi, A. Marino, N. Pisanti, R. Rizzi, G. Sacomoto *Optimal Listing of Cycles and st-Paths in Undirected Graphs*. Proceedings of SODA 2013, 1884-1896.

[9] A. Abrams and R. Ghrist, *Finding topology in a factory: configuration space*. Amer. Math. Monthly, 109, 140-150.

[10] Ki Young Ko and Hyo Won Park, *Characteristics of graph braid groups*. Discrete & Computational Geometry 48 (2012) 915-963.

[11] V. Kurlin, *Computing braid groups of graphs with applications to robot motion planning*. Homology, Homotopy and Applications, v. 14 (2012), no. 1, p. 159-180.

[12] M. Safi Samghabadi, *Collision-free motions of round robots on metric graphs*. Available at http://etheses.dur.ac.uk/7767.