Disjoint Path Algorithms for Planar Reconfiguration of Identical Vehicles

M. E. Broucke
Department of Electrical and Computer Engineering
University of Toronto, Toronto ON Canada M5S 3G4
broucke@control.utoronto.ca

1 Introduction

Suppose we have N vehicles modeled as point masses with zero width moving in the plane. Vehicle i has initial position $s_i = (x_i^0, y_i^0)$ and final position $t_i = (x_i^f, y_i^f)$. The initial and final positions satisfy a suitable general position assumption, such as all positions are distinct and no more than two positions are colinear. At time t vehicle i is at location $p_i(t) = (x_i(t), y_i(t))$. The vector of positions of all the vehicles $p = (p_1, \dots, p_N)$ is called a *configuration.* We consider the following idealized problem.

Reconfiguration problem: *Given N point mass vehicles with zero width, an initial configuration* p^0*, and a final configuration* p^f*, find disjoint paths connecting each* p_i^0 *to* p_i^f*,* $i = 1, \dots, N$*, such that the sum of the distances travelled is minimized.*

The reconfiguration problem can be interpreted in a graph theoretic way. Consider an undirected graph $G = (V, E)$, $|V| = n$, $|E| = m$. The vertices correspond to positions of the vehicles, either initial or final, or way-points along trajectories. Edges correspond to legal motions from one position to another. The edges have weights $w(e)$ representing distance or travel time between the vertices. If the initial positions of the vehicles are the vertices $S = \{s_1, \dots, s_N\} \subseteq V$ and the final positions are the vertices $T = \{t_1, \dots, t_N\} \subseteq V$, then we wish to find N vertex-disjoint paths connecting each s_i to t_i such that the sum of the weights is minimized. This problem is an instance of the minimum cost k-vertex disjoint paths problem: find k mutually vertex disjoint paths between k prespecified pairs of vertices $(s_1, t_1) \dots (s_k, t_k)$. When k can vary, that is, k is an input to the problem, and for arbitary graphs, this problem is a special case of the integral multi-commodity flow problem which belongs in the class of NP-complete problems [4]. The problem remains NP-complete in all its modes: directed/undirected graphs, vertex/edge disjoint, and is also NP-complete for planar graphs [11] and for grid graphs [8].

On the other hand, if the problem is relaxed by allowing vehicle i to go to any target location t_j, then we have a problem of finding a matching between two sets S and T. Let $G = (S, T, E)$ with $|S| = |T| = N$ be a bipartite graph with edge weights $w : E \to \mathbb{R}$. A *matching* is a subset $M \subseteq E$ such that no two edges in M share a vertex. A *perfect matching* touches all vertices exactly once, i.e. $|M| = N$. The *minimum weight matching problem* is to find a perfect matching M such that the sum of the weights of the edges in M is minimum over all possible perfect matchings. The bipartite weighted matching problem is also called the *assignment problem.* The classic solution of the assignment problem runs in time $O(N^3)$ and is called the *Hungarian Method* [9].

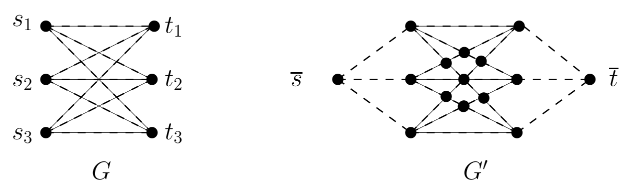

Figure 1: Graph G' for the minimum cost maximum flow problem.

The solution of the matching problem does not guarantee that vehicles follow disjoint paths. This can be done by transforming the bipartite graph G to a directed graph $G' = (V', E')$ by connecting a super source $\overline{s}$ to each s_i, connecting a super sink $\overline{t}$ to each t_i, and adding a vertex for each intersection of two paths from S to T . See Figure 1. One solves a minimum cost maximum $\overline{s} - \overline{t}$ flow problem with integer capacity constraints to ensure that paths are vertex disjoint. The min-cost max-flow solution to vehicle routing has been extensively studied in Operations Research. The approach is best suited to problems where the routes are fixed, so that potential collision points are already known. It is not a natural formulation for autonomous vehicles.

Our investigation of the problem is motivated by the following observation. Suppose the weights are derived from a Euclidean metric. Then a weighted matching algorithm always produces (vertex) disjoint paths. For, considering Figure 3(a), if two paths cross, then the targets can be swapped to yield two paths with strictly lower cost, a contradiction. In this way an algorithm can be devised with running time polynomial in the number of vehicles. Such an approach seems ideal for a large number of identical vehicles. In this paper we extend this idea to solve the reconfiguration problem using matching algorithms for vehicles moving on planar rectilinear and diagonal grid graphs. These graphs are useful when considering vehicles with nonzero width. We obtain algorithms with running time $O(N^{2.5}\log N)$, based on a suitable weighted bipartite matching algorithm [15, 16].

There is a rapidly growing literature on multivehicle problems. Graph theoretic approaches to multivehicle planning problems, in particular, have recently been considered in [1], [2], [3], [6], [10], [12], [13], and [14].

The paper is organized as follows. In section 2 we give the basic idea when distances are defined using the Euclidean metric. In section 3 we develop the matching algorithm for planar rectilinear grid graphs. In section 4 we examine the problem for planar diagonal grid graphs.

2 Planar Reconfiguration Problem

Consider a set of N vehicles with initial positions $s_i \in \mathbb{R}^2$ and final positions $t_i \in \mathbb{R}^2$, $i = 1, \ldots, N$. The notation s_i or t_i is used both to represent points in the plane and to label vertices of a graph. Let $S = \{s_1, \ldots, s_N\}$ and $T = \{t_1, \ldots, t_N\}$. We assume that the collection of points $S \cup T$ satisfy the following general position assumption: *all points in S and T are distinct and if three or more points are colinear, then they are arranged along the line in an alternating sequence of points from S and from T.* We define a weighted complete bipartite graph $G = (S, T, E)$. The edge weight $w(e)$ for $e = (s, t)$ is the Euclidean distance between s and t. Our goal is to find a matching $M \subset E$ such that:

1. The sum of the distances traveled $\sum_{e \in M} w(e)$ is minimized.
2. No straight line segments corresponding to edges included in the matching intersect. That is, for any $e = (s, t)$, $e' = (s', t')$, if $e, e' \in M$, then the segments $\overline{st}$ and $\overline{s't'}$ are disjoint. When two such segments intersect, we call it a *crossing*.

This problem is solved using a minimum cost bipartite matching algorithm taylored for the Euclidean metric, which has a running time of $O(N^{2.5}\log N)$ [16]. Results for 30 vehicles based on a linear programming solution are shown in Figure 2.

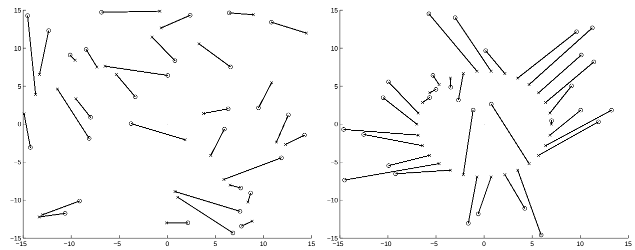

Figure 2: Disjoint Euclidean paths for random and circular arrangements of source and target positions.

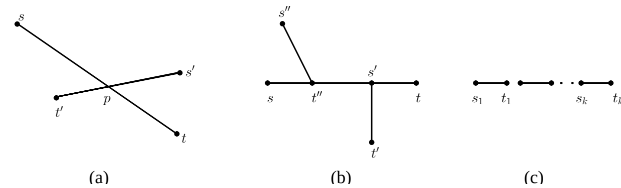

Figure 3: Tangential and transversal crossing of paths.

Theorem 1 *The solution of the minimum weight bipartite matching algorithm yields a matching with no crossings.*

Proof: Suppose the minimum weight matching M has at least two edges $e = (s, t)$ and $e' = (s', t')$ with straight line segments crossing transversally, as depicted in Figure 3(a); the intersection point p is assumed not to occur at either s,s',t, or t'. By the triangle inequality, $w((s, t')) + w((s', t)) < d(s, p) + d(p, t) + d(s', p) + d(p, t') = w(e) + w(e')$ where $d(\cdot, \cdot)$ is the Euclidean distance. The inequality is strict assuming no more than two points among s, t, s', t' are colinear. Hence if we define a matching M' obtained from M by replacing edges e and e' by (s, t') and (s', t), then M' has cost strictly less than M, a contradiction. Next we consider a tangential crossing of paths. There are two possibilities, shown in Figure 3(b). One can verify that for the right crossing, $d(s, t') + d(s', t) \leq d(s, t) + d(s', t')$ and the inequality is strict if t' is not colinear with the segment st. Similarly in the left crossing, if s'' is not colinear with st the same strict inequality results. Hence these tangential crossings cannot occur, and the only tangential crossings with segment st are colinear ones. Consider the transitive closure of tangentially intersecting paths colinear with segment st starting with the segment st (see Figure 3(c)). This set must consist of an even number of points from S and from T; otherwise at least one of them has a tangential or transversal crossing with a non-colinear point, whose possibility we have already eliminated. By the general position assumption, these even number of points must be arranged in an alternating sequence of s_i's and t_i's. By

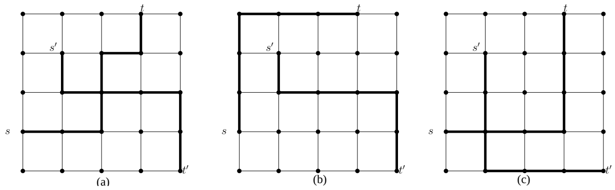

Figure 4: (a) Two Manhattan paths with a crossing. (b) The crossing is eliminated by flipping up one path. (c) Zero and one turn downward flipped Manhattan paths.

an appropriate sequence of swaps of target vertices this arrangement can always be transformed to the matching in Figure 3(c), which has a cost that is strictly less, a contradiction. Since this exhausts the possible crossings, the matching M must be cross-free. ■

3 Planar Reconfiguration on Grid Graphs

In this section we consider a weighted complete bipartite graph $G = (S, T, E)$ where the positions of the vertices lie on the integer grid in $\mathbb{R}^2$ and the edge weight $w(e)$ for $e = (s_i, t_j)$ is the Manhattan distance between s_i and t_j

$$d(s_i, t_j) = |x_i - x_j| + |y_i - y_j| \, .$$

To extend directly the idea of using minimum weight bipartite matchings to achieve cross-free reconfiguration of vehicles, the following properties must be satisfied: (1) An appropriate general position assumption holds. (2) The weights of the bipartite graph satisfy the axioms of a metric. (3) If a matching M has a crossing, that is, two paths associated with two distinct edges in M intersect, then swapping the target vertices of the edges eliminates the crossing. (4) After eliminating a crossing by swapping target vertices, the cost of the new matching is strictly lower.

The general position assumption for grid graphs we adopt is: *all points in S and T are distinct and if two or more points lie in the same horizontal or vertical line, then they are arranged along the line in an alternating sequence of points from S and from T.* The second property is satisfied because the weights are defined by the Manhattan metric. For the third property, we must verify that given any two paths associated with edges of a matching that intersect, swapping target vertices eliminates the crossing. Let $e = (s, t)$ and $e' = (s', t')$ be two edges with Manhattan paths that cross; see Figure 4(a). The intersection may be eliminated either by selecting paths of the same length but with a different number of turns, by "flipping up" or "flipping down" the paths, or by swapping target vertices. For example,

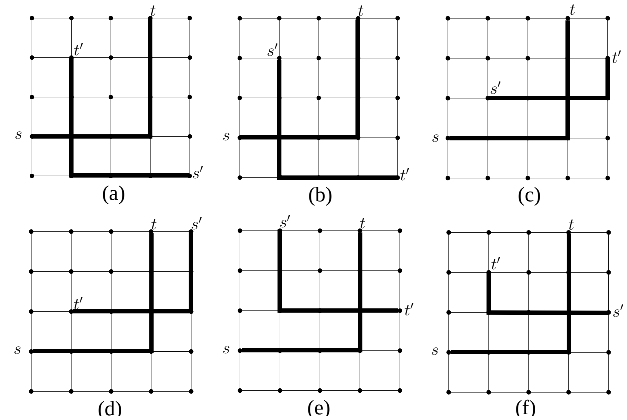

Figure 5: Types of crossings for zero and one turn downward flipped paths.

in Figure 4(b), the intersection has been eliminated by flipping up the path for (s, t). For simplicity we consider only the downward flipped paths with either zero or one turns; see Figure 4(c).

Suppose that the path for (s, t) is oriented as in Figure 4(c). The coordinates are $s = (x, y)$, $t = (x_t, y_t)$, $s' = (x', y')$, and $t' = (x'_t, y'_t)$. We consider transverse and transverse crossings with the path for (s, t) that first appear, in accordance with the general position assumption, when scanning row-wise from the top left corner of the grid. Hence, we assume that s and s' do not lie on the same horizontal or vertical line and similarly for t and t'. There are nine types of crossings with zero or one turn paths, with the six transversal ones shown in Figure 5. One can verify that swapping target vertices for these crossings yields zero and one turn downward flipped paths with no crossings.

Unfortunately property four does not hold for the Manhattan metric. One can swap two target vertices to obtain paths that do not cross with the cost equal to the cost before the swap. Considering Figure 5(c) we have $d(s, t) + d(s', t') = d(s, t') + d(s', t)$. We cannot argue as in Theorem 1 that it is a contradiction to have a crossing in the solution of the minimum weight matching problem. Moreover, if we swap target vertices to eliminate a crossing we may introduce new crossings, and it is not evident whether this process will terminate. Hence, we have

Problem: *Given M a solution of a minimum weight bipartite matching problem with weights defined by the Manhattan metric, does there exist a sequence of swaps of target vertices of pairs of edges of M whose Manhattan paths cross, such that a minimum weight cross-free matching M' is obtained?*

As before, for simplicity we consider only downward flipped zero and one turn Manhattan paths. (See [7]

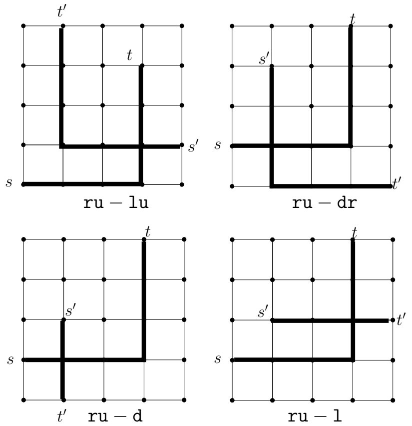

Figure 6: Infeasible transversal crossings for { `ru`, `r`, `u` }-type paths.

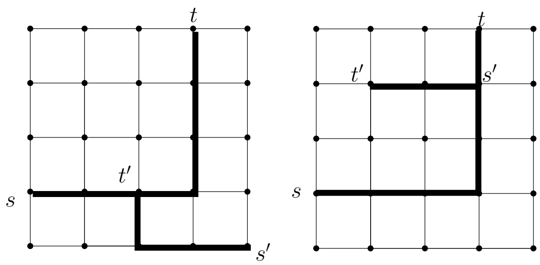

Figure 7: Infeasible tangential crossings for { `ru`, `r`, `u` }-type paths.

where a combinatorial routing problem is parametrized by the number of turns of the Manhattan paths.) We classify all such paths as right-up (`ru`), left-up (`lu`), down-right (`dr`), down-left (`dl`), up (`u`), down (`d`), right (`r`), and left (`l`).

Theorem 2 *Given M a solution of a minimum weight bipartite matching problem with weights defined by the Manhattan metric, there exists a sequence of swaps of target vertices such that a minimum weight cross-free matching M' is obtained.*

Proof: Each edge of M has a Manhattan path which is one of the types `ru`, `lu`, `dr`, `dl`, `r`, `l`, `u`, `d`. Notice that this restriction of the paths does not affect the cost. We claim that the set of path types can be organized into subsets { `ru`, `r`, `u` }, { `lu`, `l`, `u` }, { `dr`, `d`, `r` }, { `dl`, `d`, `l` }, and that crossings can only occur between path types that belong to the same subset. For instance, crossings between `ru`-`ru`, `ru`-`r`, `ru`-`u`, and `r`-`u` are feasible for the first subset. The situation for { `ru`, `r`, `u` } will be analyzed completely. The other subsets can be treated in the same way by flipping and/or rotating the grid.

Considering Figure 6 depicting possible transversal crossings between a path of type { `ru`, `r`, `u` } and a path with type `d`, `l`, `lu`, `dr`, or `dl`, we find in every case that $d(s,t') + d(s',t) < d(s,t) + d(s',t')$, contradicting that M is a minimum weight matching. The general condition is that for $e = (s,t)$ and $e' = (s',t')$ with $s = (x,y)$, $t = (x_t, y_t)$, $s' = (x', y')$, and $t' = (x'_t, y'_t)$, e and e' in M can have a crossing only if $sgn(x_t - x) = sgn(x'_t - x')$ and $sgn(y_t - y) = sgn(y'_t - y')$. Next consider in Figure 7 the tangential crossings that first appear, in accordance with the general position assumption, when scanning row-wise from the top left corner of the grid. One can verify that the path type of (s', t') cannot be `d`, `l`, `lu`, `dr`, or `dl`; otherwise the cost after swapping target vertices would be strictly less. The result is that the only feasible crossings for { `ru`, `r`, `u` } are shown in Figure 8. If the crossing occurs in row i and column j, then after swapping target vertices we have: (a) the current crossing is eliminated, (b) any new crossings resulting from the swap appear below row i and to the right of column j, (c) the new paths are of type { `ru`, `r`, `u` }, (d) if a new path of type `ru` is introduced it can only have crossings with paths of type { `ru`, `r`, `u` }, based on the arguments above, and (e) if a new path of type `r` or type `u` is introduced, it cannot have a new crossing with a path in another of the path subsets. Among the six crossing types depicted, only the last two cases can introduce new paths of type `r` or type `u` after swapping target vertices, and those new paths are segments of preexisting paths. Hence, any crossings with new `r` or `u` paths must have already existed before the swap.

We now devise a procedure to eliminate { `ru`, `r`, `u` } crossings. We start at the top left corner of the grid and scan row-wise for { `ru`, `r`, `u` } crossings. When one is encountered, we swap target vertices. Since we are scanning row-wise from the top left, only the crossing types of Figure 8 can appear, by the general position assumption. These six feasible crossing types have the property that new crossings may appear to the right of the current column, but as we scan right eventually all will be eliminated since there are only a finite number of paths touching row i. Once a row is free of crossings we proceed to the next row. Since there are a finite number of rows, this procedure terminates with an { `ru`, `r`, `u` }-cross free matching M'. This process does not add new crossings with the other path subsets. The procedure can be repeated for the other path type subsets, to obtain a cross-free matching with cost the same as that of M. ■

Remark 1 *The proof is constructive as it provides a procedure to obtain the correct sequence of swaps. The procedure can be implemented using an efficient planar sweep algorithm for line segment intersection detection. The complexity of line segment intersection reporting is $O(N \log(N) + k)$, where N is the number of vehicles and k is the number of intersections [5]. Results are*

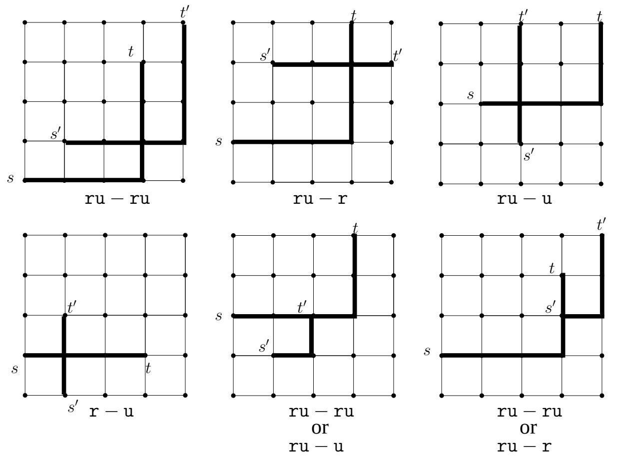

Figure 8: Feasible tangential and transversal crossings for { `ru`, `r`, `u` }-type paths.

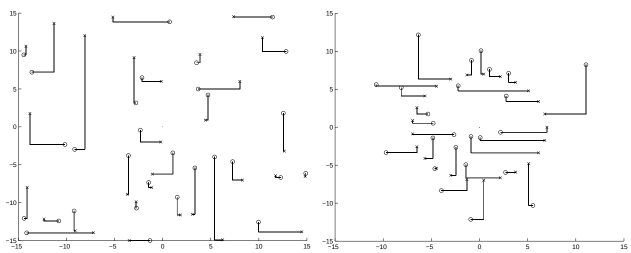

Figure 9: Disjoint Manhattan paths for random and circular arrangements of source and target positions.

depicted in Figure 9.

4 Planar Reconfiguration on Diagonal Grid Graphs

In this section we consider a weighted complete bipartite graph $G = (S, T, E)$ where the positions of the vertices lie on the integer grid in $\mathbb{R}^2$ and the edge weight $w(e)$ for $e = (s_i, t_j)$ is the diagonal Manhattan distance between s_i and t_j

$$d(s_i, t_j) = \sqrt{2}\min\{|x_i - x_j|, |y_i - y_j|\} + \big||x_i - x_j| - |y_i - y_j|\big| .$$

The same four properties as in Section 3 must be satisfied. The general position assumption for diagonal grid graphs we adopt is: *all points in S and T are distinct and if two or more points lie in the same horizontal, vertical, or diagonal line, then they are arranged in an alternating sequence of points from S and from T.* The second property is satisfied by Lemma 1. For the third property, we can verify that given any two paths associated with edges of a matching that intersect and

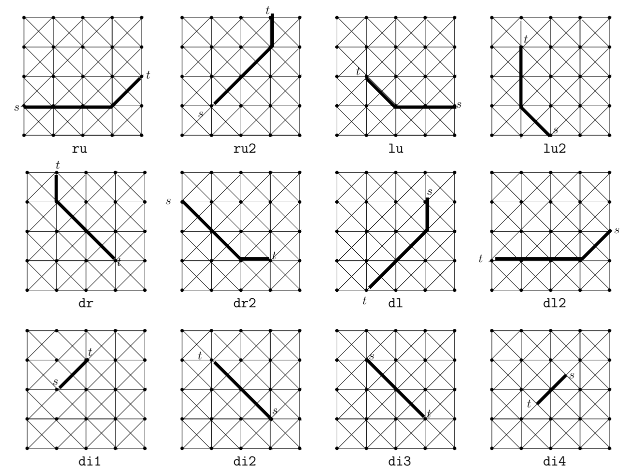

Figure 10: Types of zero and one turn downward flipped diagonal Manhattan paths.

that satisfy the general position assumption without introducing more points from S or T, swapping target vertices eliminates the crossing. We are left with property four, which also does not hold for diagonal grid graphs. We consider the following types of paths: right (`r`), left (`l`), up (`u`), down (`d`), right-up (`ru`), right-up2 (`ru2`), left-up (`lu`), left-up2 (`lu2`), down-right (`dr`), down-right2 (`dr2`), down-left (`dl`), down-left2 (`dl2`), diagonal1 (`di1`), diagonal2 (`di2`), diagonal3 (`di3`), diagonal4 (`di4`); see Figure 10.

Theorem 3 *Given M a solution of a minimum weight bipartite matching problem with weights defined by the diagonal Manhattan metric, there exists a sequence of swaps of target vertices such that a minimum weight cross-free matching M' is obtained.*

Proof: We claim that the set of path types can be organized into subsets { `ru`, `di1`, `r` }, { `ru2`, `di1`, `u` }, { `lu`, `di2`, `l` }, { `lu2`, `di2`, `u` }, { `dr`, `di3`, `d` }, { `dr2`, `di3`, `r` }, { `dl`, `di4`, `d` }, and { `dl2`, `di4`, `l` }, and that crossings can only occur between path types that belong to the same subset. As in the proof of the previous theorem, one can verify this fact by checking all possible combinations between path types. For the subset { `ru`, `di1`, `r` } the feasible crossings are shown in Figure 11. If the crossing occurs in row i and column j, then after swapping target vertices we have: (a) the current crossing is eliminated, (b) any new crossings resulting from the swap appear below row i and to the right of column j, (c) the new paths are of type { `ru`, `di1`, `r` }, (d) if a new path of type `ru` is introduced, it can only have crossings with paths of type { `ru`, `di1`, `r` }, and (e) if a new path of type `di1` or `r` is introduced, it cannot have a new crossing with a path in another of the path subsets. Indeed any crossings with new `di1` or `r` paths must have already existed before

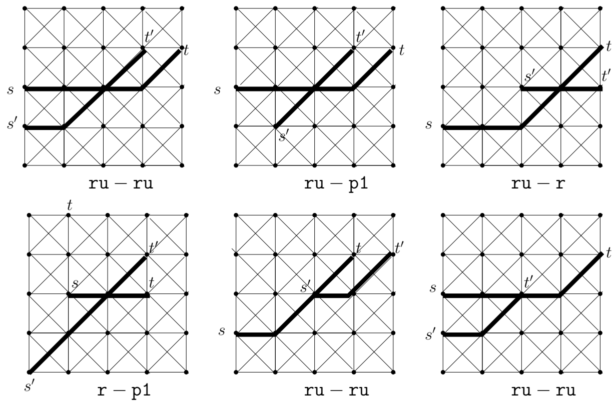

Figure 11: Feasible tangential and transversal crossings for { ru, di1, r }-type paths.

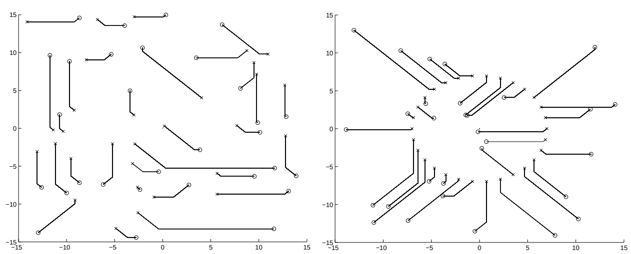

Figure 12: Disjoint diagonal Manhattan paths for random and circular arrangements of source and target positions.

the swap. Now we can devise a planar sweep starting from the top left corner of the grid, as in Theorem 1, that eliminates all crossings for { ru, di1, r } without adding new crossings with other path classes. Feasible tangential crossings for which swapping target vertices does not eliminate the crossing cannot appear in the planar sweep, by the general position assumption. The other path types are handled by the same procedure after rotating and/or flipping the graph. ■

Acknowledgements: The author thanks David Chinnery, Aniruddha Pant, Pete Seiler, and J.K. Hedrick for helpful discussions.

References

[1] M. Branicky. Nonlinear and hybrid control via RRT's. *Proc. of Mathematical Theory of Networks and Systems* (MTNS '02) May 2002.

[2] J. Desai, J. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905-908, 2001.

[3] J. Fax and R. Murray. Graph Laplacians and stabilization of vehicle formations. *Proc. of the 15th IFAC World Congress*, July 2002.

[4] M. Garey and D. Johnson. *Computers and Intractibility.* W.H. Freeman Co. 1979.

[5] J. Goodman and J. O'Rourke, eds. *Handbook of Discrete and Computational Geometry.* CRC Press, 1997.

[6] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Proc. of the IEEE Conference on Decision and Control* (CDC '02), December 2002.

[7] R. M. Karp, F.T. Leighton, R.L. Rivest, C.D. Thompson, U.V. Vazirani, and V.V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, vol. 2, pp. 113-129, 1987.

[8] M.R. Kramer and J. van Leeuwen. The complexity of wire routing and finding minimum area layouts for arbitrary VLSI circuits. VLSI Theory. F.P. Preparata, ed. JAI Press, Advances in Computing Research, vol. 2, pp.129-146, 1984.

[9] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2, pp. 83-97, 1955.

[10] N. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. *Proc. of the IEEE Conference on Decision and Control* (CDC '01), 2001.

[11] J.F. Lynch. The equivalence of theorem-proving and the interconnection problem. *ACM SIGDA Newsletter*, 5, pp. 31-65, 1975.

[12] M. Mesbahi and F.Y Hadegh, Formation flying of multiple spacecraft via graphs, matrix inequalities, and switching. *AIAA Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369-377, March 2000.

[13] R. Olfati-Saber and R. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. *Proc. of the IEEE Conference on Decision and Control* (CDC '02), December 2002.

[14] R. Olfati-Saber and R. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *Proc. of the 15th IFAC World Congress*, July 2002.

[15] J.W. Suurballe. Disjoint paths in a network. *Networks*, no. 4, pp. 125-145, 1974.

[16] P. Vaidya. Geometry helps in matching. *Siam Journal on Computing*, vol. 18, no. 6, pp. 1201-1225, December 1989.