A Resource Allocation Algorithm for Multi-Vehicle Systems with Non-Holonomic Constraints

Sivakumar Rathinam[1], Raja Sengupta[2], Swaroop Darbha[3]

Abstract— **This paper is about the allocation of tours of m targets to n vehicles. The motion of the vehicles satisfies a non-holonomic constraint, i.e., the yaw rate of the vehicle is bounded. Each target is to be visited by one and only one vehicle. Given a set of targets and the yaw rate constraints on the vehicles, the problem addressed in this paper is**

- **to assign each vehicle, a sequence of targets to visit, and**
- **to find a feasible path for each vehicle that passes through the assigned targets with a requirement that the vehicle returns to its initial position. The heading angle at each target location may not be specified.**

The objective function is to minimize the sum of the distances travelled by all the vehicles. A constant factor approximation algorithm is presented for the above resource allocation problem for both the single and the multiple vehicle case.

Note to Practitioners

The motivation for this paper stems from the need to develop resource allocation algorithms for Unmanned Aerial Vehicles (UAVs). Small autonomous UAVs are seen as ideal platforms for many applications such as searching for targets, mapping a given area, traffic surveillance, fire monitoring etc. The main advantage of using these small autonomous vehicles is that they can be used in situations where a manned mission is dangerous or not possible. Resource allocation problems naturally arise in these applications where one would want to optimally assign a given set of vehicles to the tasks at hand. The feature that differentiates these resource allocation problems from similar problems previously studied in the literature is that there are constraints on the motion of the vehicle. This paper addresses the constraint that captures the inability of a fixed wing aircraft to turn at any arbitrary yaw rate. The basic problem addressed in this paper is as follows: Given n vehicles and m targets, find a path for each vehicle satisfying yaw rate contraints such that each target is visited exactly once by a vehicle and the total distance traveled by all the vehicles is minimized. We assume that the targets are at least $2r$ apart, where r is the minimum turning radius of the vehicle. This is a reasonable assumption because the sensors on these vehicles can map or see an area whose width is at least $2r$. We give an algorithm to solve this problem by combining ideas from the traveling salesman problem and the path planning literature. We also show how these algorithms perform in the worst case scenario.

Keywords— **Cooperative control, Unmanned Aerial Vehicles, Motion Planning, Non holonomic, Travelling salesman, Dubins**

I. Introduction

THIS paper is about the allocation of tours of m targets to n vehicles. Let $(x(v_i,t), y(v_i,t), \theta(v_i,t))$ denote the position and the heading of vehicle v_i at time t. Let each vehicle start at an initial heading $\theta(v_i,0) = \alpha_i$. Similarly, let $(x(d_j,t), y(d_j,t))$ denote the position of target d_j at time t. Since the targets are assumed to be stationary, let $(\bar{x}(d_j), \bar{y}(d_j)) = (x(d_j,t), y(d_j,t))\ \forall\ t$. We assume that the Euclidean distances between any two targets and the Euclidean distance between the initial position of each vehicle and a target is greater than twice the minimum turning radius of the vehicle (the motivation for this assumption is explained later). This implies that $\sqrt{(\bar{x}(d_j)-\bar{x}(d_k))^2 + (\bar{y}(d_j)-\bar{y}(d_k))^2} \geq 2r$ and $\sqrt{(x(v_i,0)-\bar{x}(d_j))^2 + (y(v_i,0)-\bar{y}(d_j))^2} \geq 2r$, $\forall j \neq k$, $\forall j,k \in \{1,2..m\}, \forall i \in \{1,2..n\}$.

Given a set of vehicles $\{v_1, v_2, ...v_n\}$ and targets $\{d_1, d_2, ...d_m\}$, the problem is to

- assign a sequence of targets P_i to each vehicle to visit such that $\{d_1, d_2...d_m\} = \{\bigcup_i P_i\}$ and $\{P_i\} \bigcap \{P_j\} = \emptyset$ if $i \neq j$.
- assign to each vehicle v_i, a path through the sequence P_i such that the path of each vehicle v_i satisfies the following kinematic constraints:

$$\begin{aligned} \frac{dx(v_i,t)}{dt} &= v_o \cos(\theta(v_i,t)), \\ \frac{dy(v_i,t)}{dt} &= v_o \sin(\theta(v_i,t)), \\ \frac{d\theta(v_i,t)}{dt} &= \Omega \ \text{ where } \ \Omega \in [-\omega, +\omega], \end{aligned} \tag{1}$$

where, v_o denotes the speed, ω represents the bound on the yaw rate and $r = \frac{v_o}{\omega}$ is the minimum turning radius of each vehicle.

Let the sequence P_i for vehicle v_i be $d_{i_1}, ...d_{i_k}$. Assigning a path for a vehicle v_i through its sequence P_i of targets also implies assigning the angles of approach β_{d_i} at each target and assigning the angle of return β_{v_i} at which the vehicle comes back to its initial position $(x(v_i,0), y(v_i,0))$. For example, the i^{th} vehicle moves from $(x(v_i,0), y(v_i,0), \alpha_i)$ to $(\bar{x}(d_{i_1}), \bar{y}(d_{i_1}), \beta(d_{i_1}))$, and then

1. Graduate Student, Department of Civil Engg, University of California, Berkeley, CA 94720, **corresponding author** e-mail: rsiva@berkeley.edu

2. Assistant Professor, Department of Civil Engineering, University of California, Berkeley CA-94720

3. Associate Professor, Department of Mechanical Engg, Texas A & M University, College Station, TX 77843-3123.

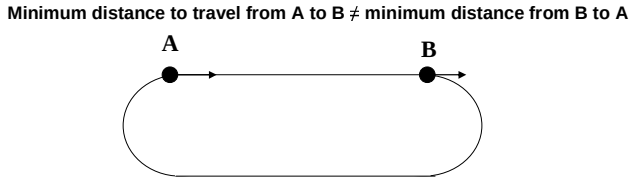

Fig. 1. An example that illustrates the asymmetry in the problem.

from $(\bar{x}(d_{i_1}), \bar{y}(d_{i_1}), \beta(d_{i_1}))$ to $(\bar{x}(d_{i_2}), \bar{y}(d_{i_2}), \beta(d_{i_2}))$ and so on. After reaching d_{i_k}, it comes back to its initial position $(x(v_i, 0), y(v_i, 0))$ at an angle β_{v_i}.

The objective is to minimize $\sum_{i=1}^{n} Cost(P_i)$, where $Cost(P_i)$ is the total distance travelled by the i^{th} vehicle.

The above problem is called as the **RAS(n)**, i.e, Resource Allocation Problem for n vehicles. If the constraints given by equations 1 are not present, then the resource allocation problem becomes a Multi-Depot, Multi-vehicle Euclidean Travelling Salesman problem. Additionally, if $n = 1$, the resource allocation problem is a single vehicle Euclidean Travelling Salesman problem.

Non-holonomy makes the costs non-Euclidean and asymmetric. Asymmetry means that the length of the optimal path of a Dubins vehicle[1] starting at target A with heading ψ_A and arriving at target B with heading ψ_B may not equal the length of its optimal path starting at target B with heading ψ_B and arriving at target A with heading ψ_A. An illustration of this asymmetry for an example is shown in figure 1.

Asymmetric Travelling Salesman Problems (ATSP) are well known in combinatorial optimization to be NP-hard [22]. Currently, there are no algorithms with a constant approximation factor available for solving ATSP problems even when the costs satisfy triangle inequality[2]. An approximation factor $\beta(P, A)$ (also called bounds) of using an algorithm A to solve the problem P (objective is minimize some cost function) is defined as

$$\beta(P, A) = \sup_{I}\left(\frac{C(I, A)}{C_o(I)}\right), \tag{2}$$

where I is a problem instance, $C(I, A)$ is the cost of the solution by applying algorithm A to the instance I and $C_o(I)$ is the cost of the optimal solution of I. In simple terms, the algorithm A produces an approximate solution to every instance I of the problem P, whose cost is within $\beta(P, A)$ times the optimal solution of I. The algorithms by Blaser [16] and Kaplan et.al. [17] for a single vehicle Asymmetric Travelling Salesman Problem problem (visiting n targets) has an approximation factor equal to $0.999 \log n$ and $0.841 \log n$ respectively. Hence, the bounds depend on the number of targets to be visited. There are also other kind of algorithms where the bound depend on the data but are independent of n. For example, the algorithm by Kumar and Li given in [18] has an approximation factor which is a increasing function of $\frac{d_{max}}{d_{min}}$. Here, $d_{max} = \max_{i,j} d(i, j)$ and $d_{min} = \min_{i,j} d(i, j)$, where $d(i, j)$ denotes the costs between two targets i and j.

RAS(n) is more general than Asymmetric Travelling Salesman problems because the heading (angle of approach) at the targets are **not** given and hence, the distances (or costs) required to travel between any two targets vary depending on the path chosen by the vehicle. We assume that the Euclidean distances between any two targets and the Euclidean distance between the initial position of each vehicle and a target is greater than twice the minimum turning radius of the vehicles. With this assumption, we present an approximation algorithm with a constant factor of 4.56 for a single vehicle and a factor of 6.07 for multiple vehicles in this paper.

A. Related work

The resource allocation problem that is addressed in this paper belongs to a class of cooperative control problems of unmanned aerial vehicles that has received significant attention in the recent years [2], [3], [4], [5], [8], [10], [13], [19]. A more general version of **RAS(n)** with multiple tasks required for each target was formulated in [15]. Task allocation and multi-assignment problems are solved using network flow and auction algorithms in [6], [7]. In [7], the number of vehicles are assumed to be greater than the number of tasks to be performed for solving the bilinear assignment problem. In this paper, we make *no assumptions* about the number of vehicles or the number of targets.

Theju et. al. in [8] present methods for solving the multi-vehicle, target assignment problem in the presence of threats with the goal of minimizing the maximum path length. In the absence of the threats, this problem is actually related to the dual of the multiple vehicle problem addressed in this paper where the sum of the distances travelled by the vehicles is minimized. Specific cases of **RAS(n)** can also be formulated as a mixed integer linear programming problem and related versions of this problem with timing constraints are tackled in [10] [11]. This approach yields an optimal solution but is computationally expensive. Since these resource allocation problems are for planning purposes, it is reasonable to devise inexpensive approximate algorithms than opt for expensive algorithms that provide optimal solutions. Also, note that there are simplying assumptions on the kinematics of the vehicle. Hence, it is logical to aim for algorithms that can provide solutions with a guaranteed approximation factor. This is the main point of this paper.

The paper that is most related to our multiple vehicle problem is the work by Zhijun et.al [19] where they provide heuristics for multiple vehicles tracking moving targets using clustering and gradient techniques. Even though [19] consider moving targets, their main results are for stationary targets which is essentially the problem that is addressed in this paper. Also heuristics for more general ver-

[1] A vehicle that satisfies yaw rate constraints.

[2] If i, j, k denote the targets to be visited and $d(i, j)$, the distance to travel from the i^{th} to the j^{th} target, then satisfying triangle inequality means that $d(i, j) \leq d(i, k) + d(k, j)$.

sions of **RAS(n)** are presented in [9] [13], but there are no bounds. Here, we present a polynomial time approximation algorithm with bounds. The approximate solution output by the multiple vehicle algorithm presented in this paper can also be used in conjunction with the heuristics presented in [19] to getter better results.

Also related to this work is the paper by Yang et. al. [14] where they consider path planning for a UAV with kinematic constraints given fixed initial and final positions in the presence of obstacles. The UAV in their work is required to visit a target and then reach a final position avoiding threats and other obstacles. This is related to the single vehicle problem addressed in this paper in the absence of obstacles when there is one target on the tour. The single vehicle problem addressed in this paper has been previously addressed by [20]. In their work, they bound the distance of the dubins path between any points (x_1, y_1, θ_1) and (x_2, y_1, θ_2) in terms of the Euclidean distance between the corresponding points. Also, using this result, they propose an algorithm which bounds the total distance travelled by the vehicle in terms of the Euclidean distance tour. In a previous paper [21] related to this work, we considered the problem **RAS(n)**, with fixed heading at each target and provided similar approximation algorithms. In this paper, we provide an algorithm for multiple vehicles with an assumption about the minimum distances between the targets.

The assumption is that the distance between any two targets and the distance between the initial position of each vehicle and a target is $\geq 2r$, where r is the minimum turning radius of the vehicle. This is a reasonable assumption in the context of unmanned aerial vehicles which carry sensors that have footprints that are greater than $2r$. For example, figure 2 shows a unmanned aerial vehicle that we use for tracking and other mapping applications. The vehicle has a minimum turning radius of around 100 meters. If the vehicle is flying at a height of at least 150 meters using a 70 degree wide angle camera, then the width of the area covered in the images is at least $2 \times 150 \times \tan(\frac{35\pi}{180}) \geq 200$ meters. Therefore, the targets within 200 meters can be seen from the same vehicle position. Hence, we have the minimum distance assumption. The algorithm given in this paper for multiple vehicles require $O((n+m)^2)$ steps where n is the number of vehicles and m is the number of targets.

II. Algorithms

In this section, we present approximation algorithms for the single vehicle resource allocation problem (i.e., when $n = 1$) and the multiple vehicle resource allocation problem ($n > 1$). Before we discuss the algorithms, we present the result by L.E. Dubins [1] which forms the motivation for the paths chosen in the algorithms.

L.E. Dubins [1] gives the optimal path the vehicle must travel between any two points subject to the path constraints given by equations 1. Henceforth, any curved segment of radius r along which the vehicle executes a clockwise (counterclockwise) rotational motion is denoted by $R(L)$, and the segment along which the vehicle travels straight is denoted by S. Thus the path in figure 3 is an RSL path. Dubin's result states that the path joining the two points (x_1, y_1, θ_1) and (x_2, y_2, θ_2) that has minimal length subject to equations 1, is one of RSR, RSL, LSR, LSL, RLR and LRL. In the special case where the points (x_1, y_1),(x_2, y_2) are at least $2r$ apart and when the angle θ_2 is not specified, the following lemma follows from the results in [1]:

Fig. 2. Unmanned aerial vehicle.

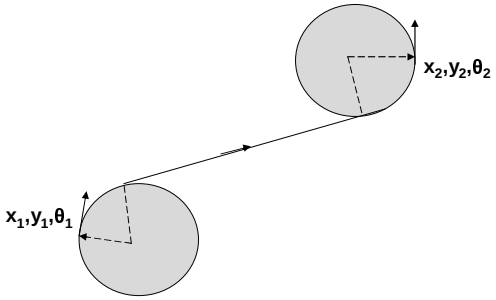

Fig. 3. Shortest path - $\{clockwise, straight, counter\ clockwise\}$.

Lemma II.1: Given (x_1, y_1, θ_1), (x_2, y_2), if $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \geq 2r$, there exists a unique RS path and a unique LS path from (x_1, y_1, θ_1) to (x_2, y_2) satisfying equations 1.

A. Single Vehicle Algorithm (SVA).

First, we give a simple algorithm **S** for the vehicle v_1 to find a path to travel from positions $(x(v_1), y(v_1), \alpha_1)$ to $(\bar{x}(d_j), \bar{y}(d_j))$. Note that the final approach angle at the position $(\bar{x}(d_j), \bar{y}(d_j))$ is free to be chosen. Algorithm **S** is as follows:

1. Find the distances of two possible paths the vehicle could take: RS and LS.
2. Choose the path that has the minimum distance.

Once, this path is followed, the vehicle reaches the position $(\bar{x}(d_j), \bar{y}(d_j))$ at some final angle θ and this angle is chosen as the heading at the final position.

The algorithm SVA to solve the single vehicle problem is as follows:

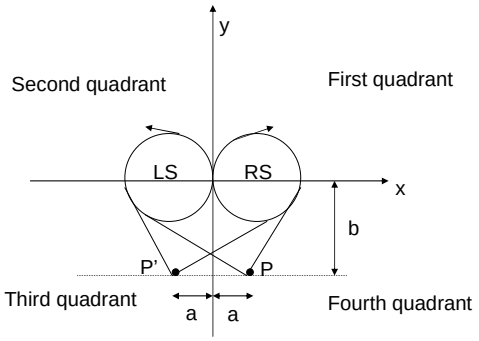

Fig. 4. The LS and RS paths.

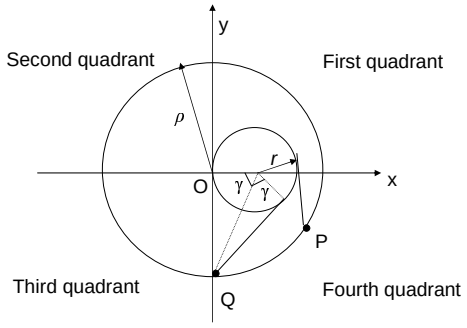

Fig. 5. Proof of lemma II.4.

1. Use Christofides algorithm [23] to solve the Euclidean Travelling Salesman problem assuming the constraints given by equations 1 are absent. The output is a sequence of targets $\{d_1, d_2, ...d_m\}$ for the vehicle to visit.
2. Use the above sequence and construct paths using algorithm **S** between any two consecutive targets. For example, use algorithm **S** to construct a path from $(x(v_1), y(v_1), \alpha_1)$ to $(\bar{x}(d_1), \bar{y}(d_1))$. Say, the vehicle reaches the target d_1 at an angle θ. Again, use algorithm **S** to construct a path from $(\bar{x}(d_1), \bar{y}(d_1), \theta)$ to $(\bar{x}(d_2), \bar{y}(d_2))$ and so on.

Lemma II.2: The path by the algorithm SVA satisfies equations 1.

Proof: Follows from the construction of the path in step 2 of the algorithm SVA. ■

A.1 Analysis

The following results provide the approximation factor for this algorithm. First, the distance travelled by the vehicle using algorithm **S** is bounded with respect to the Euclidean distance between the vehicle and the target positions. Let the vehicle be located at the origin O. Let the coordinates of the target be (x, y). Let $D(x, y)$ be the distance travelled by the vehicle using algorithm **S** to the target from O.

Lemma II.3: Without any loss of generality, let the vehicle be position at $(0, 0, \frac{\pi}{2})$ and the target be positioned at (x, y) with $\sqrt{x^2 + y^2} \geq 2r$. The path RS is optimum if $x > 0$ and the path LS is optimum if $x < 0$. Both RS and LS are optimal if $x = 0$.

Proof: Consider any two points P' and P located at $(x, y) = (-a, b)$ and $(x, y) = (a, b)$ respectively as shown in the figure 4. Points P and P' are at the same euclidean distance from the origin. As shown in the figure, the distance using the path LS to P' is less than using RS. Likewise, the distance using the path RS is optimal to reach the location P. Both the distances of the paths RS and LS are equal if $a = 0$. ■

Lemma II.4: Consider the set $S = \{(x, y) : \sqrt{x^2 + y^2} = \rho\}$. Let $\rho \geq 2r$. Then $D(x, y)$ for $(x, y) \in S$ is maximized when $x = 0, y = -\rho$.

Proof: Since the distances are symmetric about $x = 0$ axis, without loss of generality, only RS paths need be considered to targets in the first and the fourth quadrant. Let $S' = \{(x, y) : x \geq 0, \sqrt{x^2 + y^2} = \rho\}$. Consider the point Q with coordinates $(0, -\rho)$ as shown in the figure 5. Also, consider any other point P $= (x, y) \in S'$. The distances of the path RS from O to the point Q is greater than or equal to the distance to the point P. Therefore, $D(x, y)$ for $(x, y) \in S$ is maximized when $x = 0, y = -\rho$. ■

Lemma II.5: Let $\rho = \sqrt{x^2 + y^2} \geq 2r$. The ratio $\frac{D(x,y)}{\sqrt{x^2+y^2}}$ is maximized when $x = 0$ and $y = -2r$. The maximum ratio is $\pi + 1 - \tan^{-1}(2)$.

Proof: From the previous lemma, the $D(x, y)$ must be maximum for $x = 0$ and $y = -\rho$. Hence, it is enough to consider the maximization of the ratio on the set of points given by $\{(0, -\rho) : \rho \geq 2r\}$. Then from figure 5, $D(0, -\rho) = 2\pi r - 2\gamma r + \rho = (2\pi - 2\tan^{-1}\frac{\rho}{r})r + \rho$. Hence, if $\rho \geq 2r$, the ratio $\frac{D(x,y)}{\sqrt{x^2+y^2}}$ for $x = 0$ and $y = -\rho$ is $\frac{D(0,-\rho)}{\rho}$ $= (2\pi - 2\tan^{-1}\frac{\rho}{r})\frac{r}{\rho} + 1$ which is maximized at $\rho = 2r$. ■

Once the distances between the individual points are bounded, it can be combined with the Christofides result to get an approximation for the single vehicle problem.

Theorem II.1: Algorithm(SVA) solves the single vehicle problem with an approximation factor $\beta(\mathbf{RAS(1)}, SVA)$ $= \frac{3}{2}(\pi + 1 - \tan^{-1}(2)) \approx 4.56$.

Proof: The Christofides algorithm applied to the single vehicle Euclidean Travelling Salesman Problem has an approximation factor of $\frac{3}{2}$ ([23]). By Lemma $III.5$, the maximum ratio of the distance of the path constructed using algorithm **S** to the euclidian distance is $\pi+1-\tan^{-1}(2)$. Combining these two results, $\beta(\mathbf{RAS(1)}, SVA) = \frac{3}{2}(\pi + 1 - \tan^{-1}(2)) \approx 4.56$, i.e. the algorithm SVA has an approximation factor $\frac{3}{2}(\pi + 1 - \tan^{-1}(2))$. ■

Remark: The approximation factor $\beta(\mathbf{RAS(1)}, SVA)$ depends directly on the approximation factor for the Single Vehicle Euclidean Travelling Salesman Problem. This factor can be further reduced by using Arora's single vehicle algorithm [24]. Given any $\epsilon > 0$, Arora's algorithm [24] finds a solution with an approximation factor of $1 + \epsilon$ in time $n^{O(\frac{1}{\epsilon})}$.

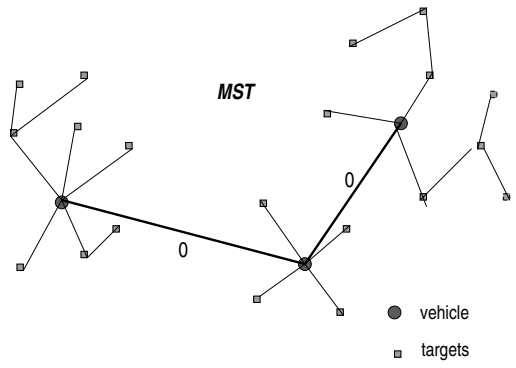

Fig. 6. Calculate the minimum spanning tree (MST). In this example, there are 3 vehicles, hence MST will have 2 zero cost edges.

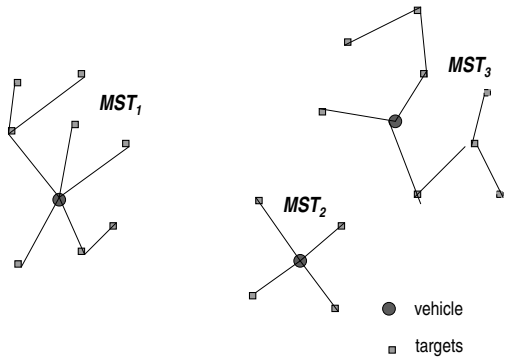

Fig. 7. Remove the zero cost edges from MST to yield MST_i for each vehicle v_i.

B. Multiple Vehicle Algorithm (MVA)

The algorithm MVA for the multi vehicle path planning problem is as follows:

1. Construct a complete graph with vertices being all the vehicles and targets. Assign the Euclidean distance as the cost to each edge that joins a vehicle to a target and a target to a target. Assign zero cost to an edge that joins any two vehicles.
2. Find the minimum spanning tree of the graph using Prim's algorithm [23]. This minimum spanning tree will contain exactly $n-1$ zero cost edges where n is the number of vehicles (figure 6).
3. Remove the zero cost edges to get a tree for each vehicle (figure 7).
4. For each tree corresponding to a vehicle, double its edges to construct a Eulerian graph(figure 8). Then construct a tour for each vehicle based on the Eulerian graph. A tour for each vehicle is a sequence of targets for it to visit (figure 9). (This step is similar to Tarjan's algorithm for a single vehicle Euclidean Travelling Salesman problem [23]).
5. Use the sequence derived from the previous step for the each vehicle and construct paths using algorithm **S** between any two consecutive targets as in the single vehicle case (figure 10).

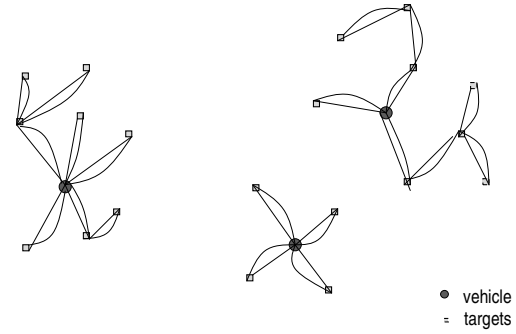

Fig. 8. After removing the zero cost edges, double the edges of the MST to get a Eulerian graph for each vehicle.

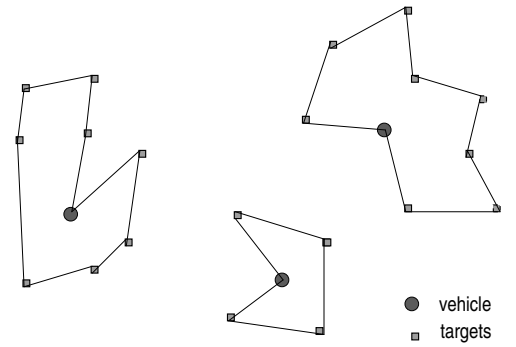

Fig. 9. Compute a tour based on the Eulerian graph for each vehicle.

B.1 Analysis

First we show algorithm MVA, without step 5, applied to **RAS(n)** without the non-holonomic constraints has an approximation factor of 2. Then, as in the single vehicle case, each edge is replaced with a path that satisfies the maximum yaw rate constraints yielding a bound similar to the single vehicle problem.

Let $G(V,E)$ be a graph with vertices $V = \{v_1, v_2 ... v_n, d_1, d_2 ... d_m\}$. The graph is complete, that is, there is a edge between every pair of vertices. Each edge is assigned a cost $C : E \longrightarrow R^+$ such that, $\forall a, b \in \{d_1, d_2 ... d_m\}$ and $\forall c, d \in \{v_1, v_2 ... v_n\}$,

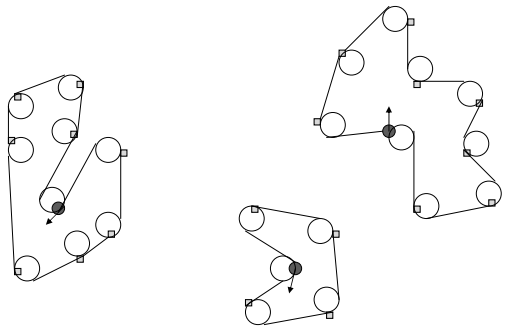

Fig. 10. Use the sequence got from the tour and construct paths using the **S** algorithm between the corresponding targets as in the single vehicle case.

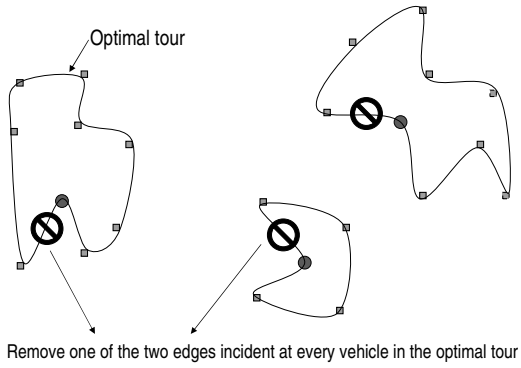

Fig. 11. Remove one of the edges incident on each vehicle in the optimal tour.

$$\begin{aligned} C(e_{ab}) &= \sqrt{(\bar{x}(a)-\bar{x}(b))^2+(\bar{y}(a)-\bar{y}(b))^2}, \\ C(e_{cb}) &= \sqrt{(x(c,0)-\bar{x}(b))^2+(y(c,0)-\bar{y}(b))^2}, \\ C(e_{bc}) &= C(e_{cb}), \\ C(e_{cd}) &= 0. \end{aligned} \tag{3}$$

Lemma II.6: The minimum spanning tree MST of the graph $G(V,E)$ computed using the Prim's algorithm has $n-1$ zero cost edges.

Proof: Prim's algorithm can be started at any arbitrary vertex in the graph G. Since, the Prim's algorithm is greedy, once a vehicle vertex is reached, it will add further $n-1$ zero costs edges before reaching any target vertex. The algorithm cannot add more than $n-1$ zero cost edges, because it would form a cycle otherwise. Hence, there will be exactly $n-1$ zero cost edges. ■

Now the following lemma gives a bound for visiting all the targets *without* considering the non-holonomic constraints given by equations 1. Let MST be the minimum spanning tree from step 2 of algorithm MVA.

Lemma II.7: Step 4 of algorithm MVA produces a sequence of targets for each vehicle to visit and has an approximation factor of 2 for visiting all the targets without the constraints given by equations 1.

Proof: Consider the optimal tours for the problem **RAS(n)** *without* the constraints given by equations 1. From each tour, remove one of the two edges that connect to the vehicle vertex to yield a tree for each vehicle as shown in the figure 11. Now, add an appropriate set of $n-1$ zero cost edges to join all the trees connected to each vehicles to make a tree connecting all the vertices $V = \{v_1, v_2 ... v_n, d_1, d_2 ... d_m\}$ (figure 12). Let this joined tree be T'. Clearly, the cost of the tour constructed using step 4 of the algorithm $MVA \geq$ the cost of the optimal tours $\geq Cost(T') \geq Cost(MST)$. Here, $Cost(T')$ and $Cost(MST)$ is the sum of the distances of all the edges in T' and MST respectively. Hence, the cost of the tour constructed using Step 4 of the MVA will always be lower bounded by $Cost(MST)$.

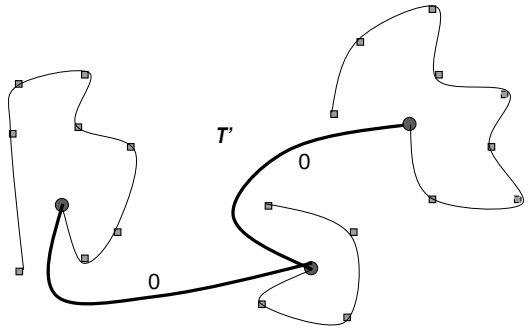

Fig. 12. Adding the zero cost edges to form a tree T' for the entire graph.

Let MST_i represent the tree for the i^{th} vehicle after removing the zero cost edges from MST (figure 7). Each tree MST_i must be a minimum spanning tree for the subset of targets connected to the corresponding vehicle. Doubling the edges results in a Eulerian graph for the corresponding subset of vertices with a cost $= 2Cost(MST_i)$. As given in [23], a Travelling Salesman tour can be constructed for each vehicle with a total cost upper bounded by $2\sum_i Cost(MST_i)$ or $2Cost(MST)$. Therefore, the cost of the tour constructed using Step 4 of the MVA will always be upper bounded by $2Cost(MST)$. Hence, the tour constructed has a bound of 2. ■

Now, the following is the result for the approximation factor of the algorithm MVA.

Theorem II.2: *Algorithm*(MVA) solves the multiple vehicle problem with an approximation factor $\beta($**RAS(n)**, $MVA) = 2(\pi + 1 - \tan^{-1}(2)) \approx 6.07$ in $O((n+m)^2)$ steps.

Proof: Follows from lemma $III.5$ and $III.7$. The time complexity is determined by three steps: finding the minimum spanning tree which takes $O((n+m)^2)$; finding the Eulerian tour which also takes $O((n+m)^2)$ steps; finding the RS, LS approximations also take $O((n+m)^2)$ steps. ■

A simulation result of the multiple vehicle algorithm with the paths travelled by each vehicle is shown in figure 13.

The positions of vehicles and targets were randomly generated in a 20 km $\times$ 20 km square with the Euclidean distances between any pair of the points $\geq 2r$. The radius of curvature r was assumed to be 0.5 km. 7 vehicles were required to visit 50 targets exactly once satisfying the curvature constraints.

C. Decentralized Implementation of Algorithm(MVA)

The decentralized implementation of the multi-vehicle algorithm (*Algorithm*(MVA)) would be useful in situations where each vehicle knows information only about a subset of targets. This would be possible if the minimum spanning tree calculation in *Algorithm*(MVA) can be implemented in a parallel or de-centralized manner. Parallel implementation of minimum spanning trees is a well-studied problem in the literature [25][26][27][28][29]. Specifically for points on a Euclidean plane, efficient algorithms are available. Since each vehicle knows information only about a subset of targets, vehicles necessarily need to communicate with each other to find a global minimum spanning tree solution. The communication strategy between the vehicles would then primarily depend on the available information. In some applications, the only information known to each vehicle could be the set of targets present in its Voronoi region. In other scenarios, maximum communication range between the vehicles can be a factor. Here, we consider one such scenario and present an algorithm where each vehicle communicates only with few other vehicles based on the work by Chang and Lee [31]. The following algorithm assumes no restriction on the communication range of the vehicles.

Let the n vehicles $\{v_1, v_2 ... v_n\}$ be arranged according to their initial abscissa such that $x_{v_1} \leq x_{v_2} .... \leq x_{v_{n-1}} \leq x_{v_n}$. Let $T(v_i)$ be the set of targets whose position information is known to vehicle v_i. A target is present in $T(v_i)$, if the abscissa of each target t in $T(v_i)$ satisfies the relation

$$\begin{aligned} x_t &\leq x_{v_i}, && \text{for } i = 1, \\ x_{v_{i-1}} \leq x_t &\leq x_{v_i} && \text{for } i = 2..(n-1), \\ x_{v_i} &\leq x_t && \text{for } i = n. \end{aligned} \tag{4}$$

Now, the implementation of the minimum spanning tree algorithm can be done in the following way:

1. Each vehicle v_i computes its minimum spanning tree for its corresponding set of nodes $S(v_i) = T(v_i) \bigcup \{v_i\}$ using Prim's algorithm. Let $i = 1$.
2. The minimum spanning trees of the vehicles v_i and v_{i+1} are merged to form the minimum spanning tree for the set of points $S(v_i) \bigcup S(v_{i+1})$ with the costs between the vehicles chosen according to equation 3. This step can be done efficiently using the algorithm given in [31]. Since the points are on a Euclidean plane, merging two spanning trees corresponding to vehicles v_i and v_{i+1} can be done in $O(|S(v_i)| + |S(v_{i+1})|)$ steps [31].
3. Let $S(v_{i+1}) := \bigcup_{j=1}^{i+1} \{T(v_j) \bigcup v_j\}$. Increment i to $i + 1$ and repeat step 2 until all the trees are merged into a single tree (i.e. until $i = n - 1$).
4. The minimal spanning tree computed at the end of the above step known to vehicles v_{n-1} and v_n is communicated to the rest of the vehicles.

This algorithm requires communication only across vehicles v_i and v_{i+1}.

III. Conclusions

This paper presented a constant factor approximation algorithm for multi-vehicle systems with non-holonomic constraints. The vehicle was modelled as a simple unicycle model with yaw rate constraints. The assumption was that the Euclidean distances between any two targets and the Euclidean distance between the initial position of each vehicle and a target is greater than twice the minimum turning radius of the vehicles. Even if the dynamics of the vehicle is included, as long as the distances travelled satisfy the triangle inequality constraints, the results given in this paper can be generalized. There are many future directions for this work. The issues that can addressed are targets with ordering constraints; targets requiring multiple visits; and stochastic uncertainty.

IV. Acknowledgements

Rathinam's and Sengupta's research was supported by ONR AINS Program - grant # N00014-03-C-0187,SPO #016671-004. Darbha's research was supported by a summer faculty fellowship at AFRL, Wright-Patterson AFB. The authors thank the anonymous reviewers for their valuable suggestions and comments.

References

[1] L.E.Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics*, vol 79, Issue 3, pp:487-516, July 1957.

[2] Phillip Chandler and Meir Pachter, Research issues in autonomous control of tactical UAVs, *American Control Conference*, pp:394-398, 1998.

[3] Phillip Chandler, Steven Rasmussen, and Meir Pachter, UAV cooperative path planning, *Proceedings of the GNC*, pp:1255-1265, 2000.

[4] Phillip Chandler and Meir Pachter, Hierarchical control of autonomous control of tactical UAVs. *Proceedings of GNC*, pp:632-642, 2001.

[5] Phillip Chandler, Steven Rasmussen, and Meir Pachter, UAV cooperative control, *American Control Conference*, 2001.

[6] Corey Schumacher, Phillip R. Chandler and Steven R. Rasmussen, Task allocation for wide area search munitions via network flow optimization, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, August 6-9, 2001.

[7] Phillip Chandler, Meir Pachter, Darba Swaroop, Jeffrey M. Fowler, Jason K. Howlett, Steven Rasmussen, Corey Schumacher and Kendall Nygard, Complexity in UAV Cooperative Control, *Proceedings of the American Control Conference*, Anchorage, AK May 8-10, 2002.

[8] Theju Maddula, Ali A. Minai and Marios M. Polycarpou, Multi-Target assignment and path planning for groups of UAVs, *S. Butenko, R. Murphey, and P. Pardalos (Eds.), Kluwer Academic Publishers*, December 4-6, 2002.

[9] R.Beard, T.Mclain, M. Goodrich and E. Anderson, Coordinated target assignment and intercept for unmanned air vehicles, *IEEE Transactions on Robotics and Automation*, 18(6), pp:911-922, December 2002.

[10] John Bellingham, M. Tillerson, Arthur Richards, J. P. How, "Multi-Task Allocation and Trajectory Design for Cooperating UAVs", *Cooperative Control: Models, Applications and Algorithms at the Conference on Coordination, Control and Optimization*, November 2001.

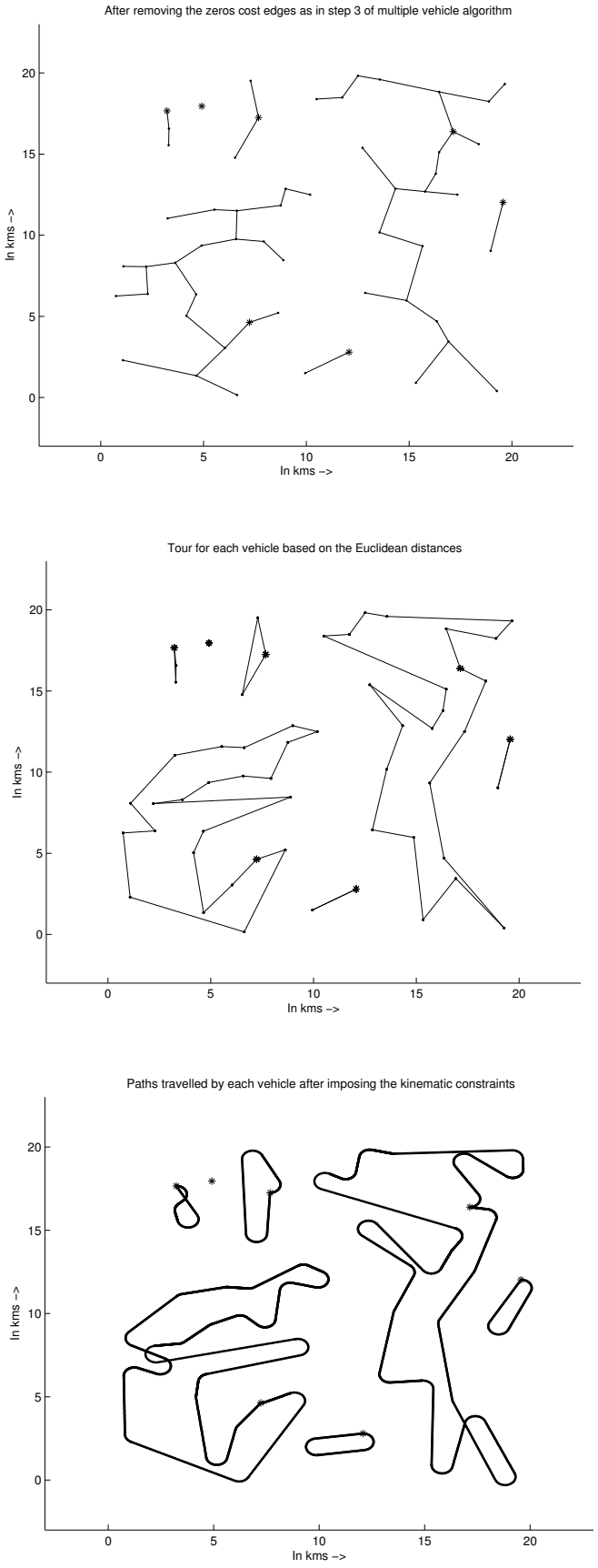

Fig. 13. A simulation result showing the application of the multiple vehicle algorithm to 7 vehicles visiting 50 targets. In the figures '*' indicates a vehicle and '.' indicates a target.

[11] Arthur Richards, John Bellingham, M. Tillerson, and J. P. How, "Co-ordination and Control of Multiple UAVs", *AIAA Guidance, Navigation, and Control Conference*, August 2002.

[12] M. Alighanbari, Y. Kuwata, and J. P. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering", *Proceeding of the IEEE American Control Conference*, June 2003.

[13] Timothy Mclain and Randal Beard, Cooperative path planning for timing critical missions, *Proceedings of the American Control Conference*, Denver, Colorado, June 4-6, 2003.

[14] Guang Yang and Kapila, V, Optimal path planning for unmanned air vehicles with kinematic and tactical constraints, , *Proceedings of the 41st IEEE Conference Decision and Control*, vol 2, pp:1301 - 1306, December 2002.

[15] Swaroop Darbha, "Teaming Strategies for a resource allocation and coordination problem in the cooperative control of UAVs", AFRL Summer Faculty Report, Dayton, Ohio, 2001.

[16] Markus Blaser, "A new approximation algorithm for the asymmetric TSP with triangle inequality", *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pp: 638 - 645, 2003.

[17] Haim Kaplan, Moshe Lewenstein, Nira Shafrir and Maxim Sviridenko, "Approximation algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs", *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pp: 56- 65, 2003.

[18] Ratnesh Kumar and Haomin Li, "On Asymmetric TSP: Transformation to Symmetric TSP and Performance Symmetric TSP and Performance Bound", Submitted to *Journal of Operations Research*.

[19] Zhijun Tang and Umit Ozguner, Motion planning for multitarget surveillance with mobile sensor agents", to appear in *IEEE Transactions of Robotics*, 2005.

[20] Ketan Savla, Emilio Frazzoli, and Francesco Bullo, On the point-to-point and traveling salesperson problems for Dubin's vehicle, *American Control Conference*, Portland, OR, June 2005.

[21] S. Rathinam, R. Sengupta and S. Darbha, Path Planning for a Collection of Vehicles with Yaw Rate Constraints, *5th International Conference on Cooperative Control and Optimization*, University of Florida, Gainesville, Florida, January 20-22, 2005.

[22] M.R. Garey and D.S. Johnson, Computers and intractability - A guide to the theory of NP-completeness, W.H. Freeman and Company, New York-San Francisco, 1979.

[23] Christos H. Papadimitriou and Ken Steiglitz, Combinatorial optimization: algorithms and complexity, Prentice-Hall 1982, Dover publications 1998.

[24] S. Arora, Polynomial-time Approximation Schemes for Euclidean TSP and other Geometric Problems, *Proceedings of the 37^{th} Annual Symposium on the Foundations of Computer Science*, pp:2-11, 1996.

[25] J. L. Bentley, A parallel algorithm for constructing minimum spanning trees, *Journal of Algorithms*, vol 1, no. 1 pp:51-59, 1980.

[26] D. Nash and S.N. Maheshwari, "Parallel algorithms for the connected components and minimal spanning trees", *Information Processing Letters*, 14(1), pp:7-11, 1982.

[27] D.B. Johnson and P. Metaxas, "A parallel algorithm for computing minimum spanning trees", In *Proc. 4th Ann. Symp. Parallel Algorithms and Architectures*, San Diego, CA, pp:363-372, 1992.

[28] F. Dehne and S. Gotz, "Practical parallel algorithms for minimum spanning trees", In *Workshop on Advances in Parallel and Distributed Systems*, pp:366-371, West Lafayette, IN, October 1998 (co-located with the 17th IEEE Symp. on Reliable Distributed Systems).

[29] S. Chung and A. Condon, "Parallel implementation of Bor uvkas minimum spanning tree algorithm", In *Proc. 10th International Parallel Processing Symposium*, pp:302-315, April 1996.

[30] David A. Bader and Guojing Cong, "A Fast, Parallel Spanning Tree Algorithm for Symmetric Multiprocessors", IPDPS 2004.

[31] Chang, R. C. and Lee, R. C. T., "An O(NlogN) Minimal Spanning Tree Algorithm for N Points in the Plane", BIT. vol 26, pp:7-16, 1986.