Coordinated UAV Target Assignment Using Distributed Tour Calculation

David H. Walker  
Brigham Young University - Provo

Timothy W. McLain  
Brigham Young University - Provo, mclain@byu.edu

See next page for additional authors

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Mechanical Engineering Commons

Original Publication Citation  

BYU ScholarsArchive Citation  
Walker, David H.; McLain, Timothy W.; and Kowlett, Jason K., "Coordinated UAV Target Assignment Using Distributed Tour Calculation" (2004). All Faculty Publications. 1527.  
https://scholarsarchive.byu.edu/facpub/1527

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen amatangelo@byu.edu.
Chapter 1

COORDINATED UAV TARGET ASSIGNMENT USING DISTRIBUTED TOUR CALCULATION

Coordinated Assignment through Tour Path Planning

David H. Walker  
Department of Mechanical Engineering  
Brigham Young University  
dhw9@email.byu.edu

Timothy W. McLain  
Department of Mechanical Engineering  
Brigham Young University  
mclain@byu.edu

Jason K. Howlett  
San Jose State University Foundation  
NASA Ames Research Center  
jhowlett@mail.arc.nasa.gov

Abstract  In this chapter a method for assigning unmanned aerial vehicle agents to targets through the use of preplanned vehicle tours is presented. Assignments are based on multi-target tours that consider the spread of the targets and the sensor capabilities of the vehicles. In this way, the individual agents and the team as a whole make better use of team resources and improve team cooperation. Planning and assignments are accomplished in reasonable computational time through the use of heuristics to reduce the problem size.

Keywords: unmanned aerial vehicles, task allocation, path planning, cooperative control
1. Introduction

A growing number of applications require the coordination and cooperation of multiple autonomous agents to accomplish a team goal. Many of these efforts utilize Unmanned Aerial Vehicles (UAVs) due to the unique capabilities they provide. In a growing number of these applications, agents must make both tactical and practical decisions autonomously. This is particularly true of systems involving teams of agents which are too complicated to be controlled or efficiently monitored by a human operator. This work applies to the coordination and cooperation of multiple autonomous fixed-wing UAVs that are subject to dynamic and sensory constraints. The vehicles cooperate in an effort to visit a number of targets and to perform a number of different tasks on those targets.

This work is relevant to the implementation of autonomous wide area search munitions (WASM). A common scenario for a WASM team is for the team to visit multiple potential targets in order to properly classify them, attack classified targets (that prove not to be decoys), and then to revisit the attacked targets to perform battle damage assessment (BDA) [Swaroop, 2000, Schumacher et al., 2002]. An example of this scenario is depicted in Figure 1.1.

![Figure 1.1. Example scenario for cooperative assignment.](image)

The general problem is to resolve who goes where and to determine how they are going to get there. These questions are subject to vehicle and problem constraints, as well as computational and timing limitations. Challenging aspects of the problem include the dynamic
Cooperative Assignment through Tour Path Planning

constraints on the individual vehicles and the overall rate of problem growth associated with multi-vehicle, multi-target assignment problems. Dynamic vehicle limitations make it difficult to plan flyable paths that make effective use of UAV sensory capabilities. Problem growth is a complication because the number of possible individual UAV tour paths and team assignments grows rapidly with increasing numbers of vehicles, targets, and tasks [Chandler et al., 2002]. This growth makes global path planning and assignment evaluation computationally intractable for problems of even modest size.

The principle issues that need to be addressed are optimal (or at least effective) path planning, target assignment, and the coupled relationship of these two tasks. In order for a team to effectively coordinate the mission plan between vehicles, it requires the management of these two coupled decision tasks. The order of execution of these tasks is not obvious due to the coupled relationship between them [Fowler, 2001, Howlett, 2001]. In order for an effective assignment to be selected it must be known how and when a vehicle will arrive at the specified target — the path or tour must be known. However, for the vehicle to plan a path it must know where it is expected to visit — the assignment must first be known.

Path planning is the process of generating a flyable trajectory that the vehicle follows in accomplishing all of its desired tasks. Planning optimal, dynamically constrained paths is a complicated nonlinear optimization problem of high degree [Reif, 1979]. There has been significant work exploring methods for effective path planning including: the use of piecewise optimal, geometrically constructed path segments and iterative assignments [Schumacher et al., 2002, Dubins, 1957]; the use of mixed-integer linear programming [Schouwenaars et al., 2001]; the use of probabilistic and random search methods [Frazzoli et al., 2002]; the construction of Voronoi diagrams [McLain et al., 2001]; the assembly of paths from preconstructed automaton path segments [Schouwenaars et al., 2003]; and the implementation of an A* path tree search [Howlett, 2002].

The majority of the methods plan paths between two fixed and known points. When paths are required that pass through multiple points, paths are generated by assembling multiple point-to-point path segments end to end. These methods guarantee path length optimality only for a given order of waypoint visits. Using conventional methods, optimal multiple-point tour paths can only be generated when the required waypoints and the order in which the waypoints will be visited has previously been determined.
The one exception to these path planning requirements is the method described by Howlett [Howlett, 2002]. This planner finds the optimal path through a series of targets while also finding the best order for visiting those targets through a Learning Real-Time A* (LRTA*) tree search. The search method used also takes advantage of the full sensing capabilities of the vehicle. By utilizing the full area of the sensor footprint, this planner produces shorter, more efficient paths. It is hypothesized that when individual UAVs plan better paths and make better use of individual UAV resources, assignments constructed from those paths will also result in improved use of team resources and increase cooperation. It is on this path planning method that this coordinated assignment work is primarily based.

The coupled problem of allocating vehicles to tasks has also received considerable attention in the literature. One method that has been used is a market driven approach in which the vehicles bid for tasks based on flight costs related to accomplishing the task [Chandler and Pachter, 2001]. Another method used to iteratively assign tasks to vehicles is accomplished through a network flow optimization model [Schumacher et al., 2002, Schumacher et al., 2003]. Others have formulated the vehicle routing problem, with various constraints and degrees of freedom, as a Mixed Integer Linear Program (MILP) [Alighanbari et al., 2003, Bellingham et al., 2003]. The problem has also been studied using gaming theory [Ganapathy and Passino, 2003]. Still other methods have been applied to ground-based robots that have relevance to the task allocation problem in UAVs [Brummit and Stentz, 1996]. The allocation methods described in these papers address some of the coupled problems of path planning and task allocation, but also often prove to be optimal only for restricted problems. These paths are often only piecewise optimal, used in situations where path planning is performed one step at a time without regard for future possible vehicle actions.

The work herein represents an alternative method for task allocation that is enabled by the use of an improved path planner. The concept is summarized in this statement: when each vehicle makes better use of individual resources through planning efficient tour paths, the team is able to improve the overall use of resources and the coordination between agents. The computationally intense path planning and combinatorially large number of assignments are managed through heuristics and estimates so that the system can produce near real-time assignments and path plans. A method using path planning developed from geometric constructions described in [Dubins, 1957] and an iterative greedy assignment method are developed and used as benchmarks for comparison.
2. **Problem Statement**

The problems to which this work applies involve systems of agents that must cooperate to accomplish a team goal. The specific problem addressed involves multiple vehicles that must cooperatively visit multiple targets. Further, each target must be visited multiple distinct times by a vehicle. The need for repeated visits to the targets arises from the distinct tasks that must be performed on the targets. Multiple visits may be required in order to properly classify a target. After classification, the target may need to be attacked and then receive a BDA sensory pass to verify that the target has been destroyed. We refer to this type of problem as a Multiple Vehicle, multiple Target, multiple Visit (MVTV) problem.

The MVTV problem described here applies to WASM which typically are fixed-wing aircraft with limited sensors that must accomplish each of the tasks mentioned above. The munitions have dynamic limitations associated with fixed-wing aircraft. The vehicles must maintain at least a minimum speed to prevent stalling, and they have a limited turning radius or maximum turning rate. For simplicity, the vehicles are assumed to fly at their maximum velocity, at a constant altitude, and are assumed to make all turns at their constant minimum turning radius.

There are a number of sensory simplifications made in this work. Each vehicle is equipped with a sensor that views the ground in a fixed position relative to the vehicle. The sensor footprint is large relative to the size of the vehicle and is placed so that it views the ground directly below the vehicle. Any target on the ground inside the sensor footprint of the vehicle is considered detected. The sensor is gimballed so that it views the ground below the vehicle whether the vehicle is in level flight or is banked in a turn. Another simplification is that the vehicles are assumed to be equally capable of accomplishing all task types. This implies that all requirements for task completion are equal to the path planner and the assignment manager, reducing the different tasks to a sequence of visits by the vehicles. A final simplification is that target positions in the area of interest are already known. This can be accomplished by a preliminary sensory pass through the area of interest by the agents resulting in a clear picture of potential targets to be visited.

A vehicle tour is a set of targets that the vehicle must visit. Problems such as the MVTV problem, in which the vehicles are subject to dynamic limitations, have the added complication of targets that are spatially coupled. The coupling is most severe when the spacing of the targets is on the order of the turning radius of the vehicles. Coupling between path
segments is apparent whenever a path segment concludes in a heading that prevents the vehicle from readily accomplishing a subsequent visit.

Many path planning methods are based on point-to-point optimal planning. The benchmark path planning method that is used for comparison of results is such a planner and is used in [Schumacher et al., 2002, Chandler and Pachter, 2001, Schumacher et al., 2003]. This planner is based on the mathematical work of L.E. Dubins [Dubins, 1957]. In a point-to-point planner the initial and final positions and headings of a given flight segment influence the optimal path for the segment. When a path is required to pass through multiple points, the points to be visited and the order in which they are to be visited must be specified to the planner. The point-to-point path planner designates the position and heading of the vehicle at the completion of a path segment, and thereby also fixes the initial position and heading of the vehicle for the subsequent path segment. Spatial coupling occurs because the route to a subsequent target depends heavily on how previous visits were completed. Path planners that find an optimal path for a given sequence of positions and headings may not obtain the optimal trajectory simply because the sequence of waypoints was not optimal. Even when the sequence is optimal, and each of the point-to-point segments are optimal, the resulting multi-target path may be significantly longer than necessary due to this spatial coupling and incorrect selection of vehicle headings at the completion of each task. A case illustrating how this can happen is shown in Figure 1.2.

![Figure 1.2. Coupling between path segments.](image)

(a) Correct sequence may yield suboptimal paths.  
(b) Optimal multi-target path plan needs correct sequence and headings.
An effective solution for MVTV problems requires an improved trajectory planner and an efficient method of assignment selection that is capable of managing problem growth and meeting computational speed requirements. The planner should

- plan optimal or near-optimal tour paths for closely spaced targets
- make full use of the entire sensor footprint
- plan complete tours over multiple targets, some requiring multiple visits
- find the best tour without specification of tour visit order.

The planner utilized here determines the best path through a given set of targets, including the tour order and the optimal multi-target path. This trajectory planning method will be described in greater detail in Section 3.1.

As discussed earlier, the coupling between path planning and target allocation is a significant issue in MVTV problems. The dilemma is that an assignment is needed for the vehicle to plan a path, but the details of the path are required for effective team assignments to be made. A common approach to overcome this dilemma is to plan path segments and make single assignments iteratively. The vehicles plan optimal path segments from their current location to the various targets that need immediate attention. Greedy assignments are then made based on the costs for the vehicles to accomplish the immediate tasks. The problem that arises is that the assignments and planned paths make no consideration for the state of the system at the conclusion of the various tasks. The vehicles often complete the present task in an optimal manner, but are in poor condition to address subsequent unfinished tasks. Furthermore, iterative methods may lead to “churning” in the assignment. Churning occurs when a vehicle is assigned to a task, but is later unassigned on a subsequent assignment iteration because it is determined that another vehicle will be able to accomplish the task first. Iterative assignment methods, although fast, often lend themselves to overall system inefficiencies, lengthy paths, and poor cooperation among the agents because the assignment is myopic with no concern for future actions.

An improved planner that results in better tour paths can be used to improve assignments. In selecting assignments the managing algorithm must take a number of factors into consideration. The assignment algorithm should

- efficiently setup the problem — find complete assignments and possible UAV tours
utilize paths planned by the individual vehicles’ tour planners

effectively manage problem growth issues

efficiently evaluate assignment costs, returning good assignments in reasonable time.

For the MVTV problem, increases in the number of cooperating vehicles, the number of targets, and the number of required visits to each target result in explosive growth in the number of possible tours and team assignments. This growth in problem size affects the computational requirements for both the tour planner and the algorithms used in assignment setup and selection. As a result of this explosive growth, viable methods must focus on the development of fast algorithms and methods for reducing the problem size.

The objective of this work is to improve team cooperation through improved tour paths. A tour planner creates optimal tour options for each UAV without a priori knowledge of tour order. Assignments are then selected by combining appropriate tours from the separate UAVs. These assignments and paths fulfill the global team goal rather than looking only one step ahead, and improve use of team resources and overall cooperation between the agents.

3. Technical Approach

The approach presented here achieves the goals set forth through the use of an improved path planner for individual flight tours coupled with an efficient approach for task management. The calculation of a tour path allows the consideration of the overall benefit of an entire team assignment, rather than iteratively evaluating the immediate gain of individual vehicle subassignments.

The path planner uses a learning algorithm that makes it capable of accomplishing the various required tasks. The planner is described in Section 3.1, defining how it works and its limitations. The assignment algorithm is presented in Section 3.2. Various aspects of the assignment process are described. First, the problem setup and the utilization of the tour path planner are explained. Methods for controlling problem growth in the assignment process are then discussed. The overall algorithm is presented in Section 3.3, illustrating how the computation can be distributed across multiple computers to further manage the computational load.
3.1. Tour Path Planner

The tour path planner developed in [Howlett, 2002] implements a discrete-step path planner to search a tree of possible paths. The goal of this path-planning approach is to find the branches of the tree that result in the agent meeting the objectives set forth. Once a set of branches have proven to meet the objectives of the planner, the shortest branch is selected as the planned path. Due to the well-defined nature of the discrete tree, it lends itself to a Learning Real-Time A* (or LRTA*) search to explore the tree for branches that meet the desired objectives. The specific implementation of the LRTA* algorithm developed by Howlett is unique because there is no set goal node. The objective is met as the path weaves its way through the spatially close targets and is able to sense each of them. In the original work [Howlett, 2002], the objective was to sense the multiple targets only a single time each. The algorithms, heuristics, and path goals have been modified in this work to allow multiple repeated visits to individual targets as required by the MVTV problem definition.

An example of an LRTA* path tree is depicted in Figure 1.3. The tree is constructed of left-turn, straight, and right-turn segments of discrete length. The root of the tree is at the initial vehicle location. The tree is constructed so that the branches span the area of interest.

![Figure 1.3. Primitive turn and straight path segments of equal length, $dS$, are assembled to form a tree of flyable paths.](image)

The LRTA* algorithm is actually quite simple and proceeds in the following manner [Weiss, 2000]. Each discrete-step node, $i$, has a heuristic, $h_i$, which estimates the path length to be travelled by the vehicle before the multiple-target sensing objective is accomplished. Every node has a set of $m$ neighbor nodes, which are the discrete-step nodes that the vehicle can proceed to next. At each step of the search, the current
node, $i$, calculates

\[ f_j = k_{ij} + h_j \quad \forall j = 1, \ldots, m. \]  

(1)

The value of $k_{ij}$ is the cost for the vehicle to travel from node $i$ to node $j$. The value of $f_j$ is the estimated path length before the objective is met if the vehicle at node $i$ proceeds to neighbor $j$. The node $i$ heuristic is updated so that $h_i = \min_j f_j$, and then the algorithm proceeds to the corresponding minimum cost neighbor. The algorithm proceeds from node to node in this manner, updating the heuristics as it goes, until the objective is reached (all targets sensed), and the search is begun again at the initial node. For each step of the search, the heuristic for the current node is updated with a better estimate until the updates converge to the actual minimum path.

There are two major issues for consideration when initializing heuristics in the LRTA* planner. The fundamental requirement of the LRTA* path search method is that the individual node path heuristics must always initially underestimate the true path length. This heuristic admissibility restriction is required by the algorithm because if it initially overestimates the path length then the algorithm may never explore branches of the discrete tree that actually lead to the optimal solution. The second issue that is pertinent to the effectiveness of LRTA* search is the initial value of the heuristic. The closer the initial heuristics are to the actual path length value, the faster the algorithm will converge to the optimal path.

The learning algorithm that is used is actually a non-improving version of the LRTA* algorithm. The Non-Improving LRTA* (or NILRTA*) is identical to the general LRTA* algorithm except that it has an additional search terminating condition. The LRTA* algorithm only terminates when the heuristics along the optimal path have converged to the actual path length value. The LRTA* algorithm quickly finds optimal or near-optimal paths, but spends most of the computation time either tweaking the path for minor improvement or simply verifying that the path found is optimal. The NILRTA* algorithm, described in [Howlett, 2002] and used in this work, uses a search terminating condition in addition to the heuristic convergence used in LRTA*. When the algorithm has gone through a given number of iterations without finding a better path, the algorithm terminates and returns the best current path. In this way the algorithm is able to trade off minor improvements in path planning performance for major gains in speed of the computation. Two sample paths for the same multi-target tour are shown in Figures 1.4(a) and 1.4(b). The tour in Figure 1.4(a) represents a sample path from a point-to-point planner. Figure 1.4(b) illustrates a tour planned using
the NILRTA* tour planner. The case shown, the tour-planned path is only 41 percent as long as the point-to-point path. The tour-planned path is capable of completing the identical tour in significantly less time due to the effective use of the entire sensor footprint enabled by the NILRTA* planner.

Figure 1.4. Sample paths generated using a point-to-point planner and NILRTA* tour path planner.

3.2. Team Assignment Strategy

A large portion of the assignment problem is simply tied up in generating assignments that are both complete and not redundant. An assignment is complete when every target is fully serviced by the UAV team. A redundant assignment is one in which more visits are made to a given target than are required. Assignments cannot make effective use of team resources if they either fail to service the targets or if they are assigned to over-service certain targets. An iterative approach that has been used in [Schumacher et al., 2002, Schumacher et al., 2003] guarantees complete assignments that are not redundant. However, the iterative approach can result in assignments and paths that are short-sighted in scope and objective, and can often result in a less effective use of team resources. When the vehicles plan paths through an entire tour they make better use of resources that result in better team assignments. This is the objective of the method presented here.

The problem is setup in a manner that produces only complete and non-redundant assignments for the vehicles on the team. The first step taken in generating a complete assignment is to first make a list of all
possible ways that each target can be visited. For instance, a target that
must be visited three times by a team of three vehicles can be visited in
the combinations shown in Table 1.1. The way the data is presented, the

Table 1.1. The ten possible combinations of three UAVs that can be assigned to visit
a three-visit target. Assignment 2 results in vehicle 1 visiting the target twice and
vehicle 2 visiting the target once.

<table>
<thead>
<tr>
<th>assignment number</th>
<th>assigned vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>2</td>
<td>1 1 2</td>
</tr>
<tr>
<td>3</td>
<td>1 1 3</td>
</tr>
<tr>
<td>4</td>
<td>1 2 2</td>
</tr>
<tr>
<td>5</td>
<td>1 2 3</td>
</tr>
<tr>
<td>6</td>
<td>1 3 3</td>
</tr>
<tr>
<td>7</td>
<td>2 2 2</td>
</tr>
<tr>
<td>8</td>
<td>2 2 3</td>
</tr>
<tr>
<td>9</td>
<td>2 3 3</td>
</tr>
<tr>
<td>10</td>
<td>3 3 3</td>
</tr>
</tbody>
</table>

assignments (1 2 1) and (2 1 1) are identical to the shown assignment (1
1 2), and therefore are not listed. This is because the planner finds the
best order to accomplish the three tasks and does not need to be told
explicitly.

The number of possible vehicle combinations for servicing the $i$th target,
$T_i$ (ten in the case illustrated in Table 1.1), is a function of the number
of visits the target requires, $n_i$, and the number of vehicles on the team
that are used in the assignment, $m$, and is given by the relationship

$$T_i = \frac{((m - 1) + n_i)!}{(m - 1)! \cdot n_i!}. \quad (2)$$

The complete and non-redundant assignments are obtained from all
the possible combinations of the individual target service combinations.
When multiple targets are involved, the total number of possible assign-
ments, $A$, is obtained from the product of all the $T_i$’s from the individual
vehicle visit combinations for each target

$$A = \Pi_{i=1,2,...,I} T_i. \quad (3)$$

Making assignments in this manner will always result in a complete
assignment that will service all targets without redundancies.

Figure 1.5 illustrates the combinatorial growth that occurs in MVTV
problems. The growth data presented involves targets that must each be
visited three distinct times. The total number of possible assignments make it computationally intractable to perform exhaustive searches to find global solutions in near real-time applications.

Path length heuristics and team cost estimates are used to quickly approximate the cost or value of a given assignment without actually planning the paths. The large number of tours in MVTV problems makes it impractical to plan all paths with a NILRTA* path planner for global solutions within reasonable time constraints. As a result, the assignment algorithm requires more simplistic approximations of path length in order to get preliminary estimates of assignment costs and benefits. These initial approximations are used to prune obviously poor vehicle tour paths and team assignments from consideration so that computational time and effort is not further wasted planning or evaluating them. This is a necessary step to get the near real-time response that is desired.

The length of each individual tour path for each of the vehicles is approximated using a functional relationship rather than a learning search. In estimating the length of a path, the function considers the spread of the targets (the distance between the two targets furthest from each other), the number of visits required by each target, the spatial position of each target with respect to other targets in the group, and the size

![Figure 1.5](image-url)
and orientation of the UAV sensor footprint relative to the vehicle flight path.

The individual path heuristic costs are combined to get estimates for entire team assignments. The cost of an assignment is estimated by combining tour heuristics from the several vehicles as though the heuristics were the actual path lengths of the complete tours. The assignment cost estimates allow the assignments to be ordered according to their approximate relative value. The ordered list gives the priority for planning and evaluating the actual paths and assignments.

The ordered list is also used to reduce the number of assignments and paths under consideration. After the assignments have been ordered, only the \( N \) best assignments are kept for actual evaluation. The value of \( N \) is determined by the problem size and is used to control problem growth. Effective control of problem growth through the choice of \( N \) is demonstrated in Section 4.

### 3.3. Algorithm

Computations associated with path planning and assignment can be broken into portions that are either centralized or distributed. MVTV problems, by definition, are composed of multiple distinct agents that work together. The ability to manage problem growth can be improved by distributing the computational burden. The computational load is distributed to each of the individual UAV agents for path planning, and to a managing agent for problem setup, information management, and assignment evaluation. The assignment manager can be an additional computer agent in the lead UAV, or it can be in a separate agent in a command center location — possibly in a nearby ground station or in a high flying UAV.

The calculation of the individual vehicle path length heuristics is initially performed by both the assignment manager and the individual UAV agents. The heuristic calculations execute fast enough that it is simpler, more robust, and requires less communication to have every agent perform this initial estimation independently. The individual agents calculate the path length heuristics for all the tours they can conceivably be asked to perform. As the heuristics are calculated, each UAV does a preliminary ordering of tour paths based on their potential benefit. The individual UAV agents do not have the benefit of knowing how their tour will fit in with the rest of the team, but they are able to determine whether or not the tour effectively uses their individual resources. While the UAV agents are awaiting further instructions from the assignment manager, they continually calculate actual tour paths in
Cooperative Assignment through Tour Path Planning

the order of this initial ordering. In this way, the agents waste no time waiting, and instead perform calculations that they deem most useful to the team.

The managing agent is responsible for initial problem setup as well as the preliminary estimation and ordering of path heuristics and team assignments. The assignment manager calculates tour path heuristics for every tour of every vehicle in the team and then assembles team cost estimates by combining tour heuristics from the several vehicles. As the estimates are calculated, they are also ordered by estimated cost. The manager uses the ordered assignment estimates to initially reduce the size of the problem under consideration by keeping only the $N$ best assignments based on estimated costs. The ordered list of team assignment estimates and the associated tours of each vehicle are then communicated to the individual UAVs for calculation.

Upon receiving a list of tour paths from the manager, each vehicle will have a limited number of potential tour paths present in the top $N$ ordered team assignments. It is only these tours that the individual UAVs must calculate with the NILRTA* tour planning method. The UAVs plan their own individual tours in the order they appear in the ordered list of team assignment cost estimates obtained from the manager. Once a vehicle has planned a NILRTA* discrete path, the resulting path is immediately communicated to the managing agent for evaluation.

As new tour path data comes into the manager, the tour costs are combined and actual assignment costs are determined. A team assignment is then ordered on a separate list based on the actual cost of the assignment. The best assignment yet evaluated will always be at the beginning of the ordered list, ready for execution should a valid assignment be immediately required. This method can return a valid, executable solution at any time. In this way, the algorithm lends itself to situations where the planning times out, requiring a ready solution to be executed immediately.

Figure 1.6 gives an overview of the algorithm and shows the separate distributed and centralized aspects of the computation and is presented in Figure 1.6. First, the managing agent is responsible for problem setup and initialization. Similarly, the central manager is responsible for prioritizing the calculation of team assignments and individual vehicle tours. In a fully distributed manner, the UAV agents are then responsible for calculating their own individual NILRTA* discrete-step tour paths. After the tours have been calculated, the results are communicated to the managing agent for centralized evaluation and team assignment selection.
Assignment Manager Agent (Central Calculations)

Problem Setup Tasks
- Select value for $N$
- Generate list of all possible tours and assignments
- Calculate path heuristics for all tours, all vehicles
- Calculate estimates of all assignment costs
- Order $N$-best assignments according to cost estimates
- For each vehicle:
  - Extract tours from $N$-best ordered list
  - Communicate ordered tour list to each to vehicle

Assignment Evaluation Tasks
- Performed when tour costs arrive from vehicle agents
  - As actual tour cost data arrives from vehicle agents
  - Evaluate assignments for which all tour data has arrived
  - Order evaluated assignments
- Once all $N$-best assignments are evaluated
  - OR system times out
  - Return best calculated assignment
  - Communicate final tour assignments to vehicles

UAV Agents (Distributed Calculations)

Vehicle Agent 0

Vehicle Agent $i$

Setup Tasks
- Generate list of possible tours
- Calculate all tour path heuristics for vehicle $i$
- Order tours based on effective resource use

Path Planning Tasks
- Performed when setup completes or manager tours are received
  - While waiting for manager tour list:
    - Calculate NILRTA* paths for tours on local ordered list
    - Communicate tour costs to manager
  - Upon receiving manager tour list:
    - Calculate NILRTA* paths for tours not already calculated
    - Communicate tour costs to manager

Figure 1.6. Tour planning and assignment selection algorithms.
There are a number of factors that contribute to the speed of the algorithms and the overall method, and each requires individual tuning to maximize speed without reducing the quality of the result. The factors listed below have been tuned for best results in speed and quality:

- number of nodes in the discrete planner — determined by the world dimensions and the size of the discrete step.
- limits on the number of tasks a UAV can perform in a tour and on the length of the tour path.
- number of iterations of the Non-Improving planner before the search times out.
- limits on the total problem size, in number of assignments that must be setup — a function of the number of vehicles, targets, and the number of visits needed to each target.
- number of assignments kept by manager in the $N$-best assumption.

The values to which these factors are tuned depend on computer resources that are available to both the assignment manager and the individual UAVs. It is particularly important that the individual UAV agents have sufficient computer memory for calculation of the NIL-RTA* discrete paths. The world dimensions and planner step size are limited by the memory available to the UAV agent. Though memory is an issue for the assignment manager in extremely large problems, the speed of the processor is of much greater importance for this agent.

4. Results and Discussion

In this section the results of using the tour path planning method and a team assignment methodology are compared to established methods. The baseline method that is used for comparison uses a point-to-point path planner similar to the planner developed by the AFRL/VACA [Schumacher et al., 2002, Chandler and Pachter, 2001, Schumacher et al., 2003], which is based on the geometric study of L.E. Dubins [Dubins, 1957]. The baseline method also uses an assignment method that is iterative and greedy. The greedy method is used to compare myopic, iterative results with those obtained using tour paths and overall team assignments.

The greedy and myopic methods used here are straightforward implementations similar to existing iterative assignment and segment-optimal point-to-point path planning methods. The results and successes of the method are similar to those reported in previous works [Schumacher
et al., 2002, Schumacher et al., 2003]. These types of methods are useful in dealing with large problems since only a small portion of the problem is considered at any one time. The result is that large problems are automatically reduced in size and are evaluated in computationally tractable pieces.

Although computationally efficient, applications of iterative methods often result in team assignment inefficiencies and lengthy vehicle paths. A typical assignment that demonstrates this can be seen in Figure 1.7. The iterative assignment is created by determining which vehicles can visit a target most immediately. A target visit is accomplished by flying directly over the target and makes no additional use of the sensor footprint. The example shows that vehicles passed close enough to targets to have them within their sensor footprint, but the vehicle was not expecting it and had to return and fly directly over it. The result of many assignments such as this is that multiple vehicles are used to accomplish what a single UAV could do. Churning in the assignment is evident in this example. The vehicle represented by the star waypoint path made its final turn to return to a target that was reassigned to another vehicle just before the diamond UAV could complete the task. Even though one vehicle was able to visit two targets in quick succession, it still took longer than would have been necessary if the vehicle had been able to utilize the full sensor footprint.

The same scenario as was used in Figure 1.7 was run using the tour plan assignment method to compare resulting assignments. In contrast to the inefficient assignment and lengthy tours obtained with greedy, iterative methods, the method presented here results in shorter individual tours, better team cooperation, and as a result, faster overall completion of the team goal. The assignment obtained from the tour plan assignment method is presented in Figure 1.8. The use of planned tour paths results in tours that accomplish more in less time through the effective use of the entire sensor footprint and better overall cooperation. The tour-planned paths in this case result in an assignment that is completed in approximately half the time required to complete the iterative greedy tour.

4.1. Method Comparisons

Iterative assignments can lead to poor use of vehicle and team resources. The proposed method overcomes these weaknesses through better individual UAV tour planning and overall team assignments. The approach used here plans for both immediate and future target visits. When cooperating UAVs plan multi-target tours and make assignments
Figure 1.7. A sample assignment reached through execution of a greedy and iterative assignment method that employs segment-optimal path planning. Dimensions in feet.
based on these tours, the team can better utilize the mission capabilities of the individual UAVs.

Over 120 randomized tests were performed in an effort to quantify the difference between the benchmark method and the approach discussed here. Each test involved randomizing the following parameters:

- number of vehicles in the scenario — between 2 and 5
- starting UAV positions and headings — anywhere within 9000 ft. by 9000 ft. area
- number of targets to be visited — between 2 and 4
- target positions — anywhere within an 8000 ft. by 8000 ft. area.
In the randomized tests, the approach of planning tours and using those tours in making assignments proved to produce significantly better tour paths and team assignments than did the iterative and piecewise benchmark method. On average, the iterative assignment method produces tours that are 78 percent longer in time to completion than the tour-planning method proposed here. In multiple cases, the iterative assignment produced an assignment that was over four times longer than the tour-based solution. At the other extreme, the iterative approach produced better solutions in some cases, with the greatest benefit being an assignment completion time that was 11 percent shorter. These results are summarized in Table 1.2.

Table 1.2. Iterative assignment costs compared with tour-based approach costs.

<table>
<thead>
<tr>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average:</td>
<td>iterative cost = 1.78 × tour-based cost</td>
</tr>
<tr>
<td>Maximum:</td>
<td>iterative cost = 4.22 × tour-based cost</td>
</tr>
<tr>
<td>Minimum:</td>
<td>iterative cost = 0.89 × tour-based cost</td>
</tr>
</tbody>
</table>

Situations in which our method obtains the greatest improvement in overall team cooperation are exactly where previous methods have obtained the most undesirable results. Point-to-point planners are weakest when the Euclidean distance between any two targets in the tour is less than twice the turning radius of the UAVs. The complication from target spread proximity is compounded in MVTV problems when multiple visits are required by each target. In these cases the vehicles stand to gain the most from the effective use of the full sensor footprint, something that point-to-point planners are not capable of providing.

There are occasional situations in which the point-to-point planners actually produce team assignments that are better than those produced with the tour-based method. Though uncommon, these assignments generally happen when the spread of the targets is greater than twice the turning radius of the UAV. Significant spread in the targets affects the efficiency of the proposed method in two ways. First, the effective use of the UAV sensor footprint becomes less important. Second, the larger the spread of the targets, the larger world size becomes, which results in a larger number of nodes in the discrete planner. This can degrade assignments since longer paths may cause the discrete planner to time out and return a suboptimal tour. In situations where multiple ineffective tours are returned, the resulting assignments can actually be worse than the point-to-point path planning and iterative assignments of the benchmark method. This occurs primarily due to poor tuning
of the algorithm parameters for the specific problem. The parameters were tuned for general results, and the same parameters were used for all test problems. If the tuning were modified for the specific needs of problems with larger target spread, optimal or near-optimal tours and assignments would be obtained.

4.2. Reducing Problem Size

The average improvement of a team assignment using the proposed approach is considerable, but does not give a complete picture of the value or cost of the approach. Assignment benefits include faster completion time of the team assignment, improved UAV cooperation, better use of vehicle sensors and resources, and an improved ability to visit and service spatially close targets. However, even with these gains, if the approach is to be useful, the results need to be obtained within reasonable time limits and with reasonable computational resources.

A necessary part of ensuring that the problem remains computationally tractable is reducing the size of the MVTV problem space that is explored for the selection of a final assignment. MVTV problem reduction is possible through the use of tour path length heuristics and estimations of team assignment costs. In this way, the team can weed out obviously poor paths and assignments so they will not need to be fully planned and evaluated. Iterative methods control problem size by only considering a portion of the total assignment at a time, while the tour planning assignment method controls problem size through efficient elimination of tours and assignments that are unlikely to produce good results.

MVTV problems can be effectively reduced due to the nature of the assignment cost estimates generated from the heuristics. Each of the 120+ scenarios tested were solved globally while also maintaining a record of the ordered heuristics. In this way, reduced problem solutions and ordered heuristics can be compared directly to the global solution and actual ordered costs, and thereby determine the effect of maintaining only a fraction of the potential assignments in the $N$-best assignments assumption. Figure 1.9 illustrates the average position of the actual global solution on the list of assignments ordered by the heuristic cost estimate for problems of various sizes. The chart suggests an average value for $N$ to be used in problems of different sizes when a high probability of finding the optimal or global solution is desired. As can be seen, the percentage of assignments improperly ordered above the global optimum decreases as the problem size increases.
Although the percentage of team assignments that must be maintained in the $N$-best assumption decreases as the problem gets larger, the total number of assignments and tour paths calculated still increase. The result is an upper limit on problem size that is governed by computer speed and by the desired quality of the result. The value of $N$ depends on the problem size. For problems with fewer than 1000 possible assignments, globally optimal solutions could be reliably found by fully computing only the top 20 percent of those assignments. For problems with more than 1000 possible assignments only the top ten percent would need to be computed to reliably find the global optimum. Improved accuracy of tour path length heuristics and team assignment cost estimates would result in a better initial ordering of assignments and a reduction in the percentage of the total assignments that would need to be included in the $N$-best path list. However, if the value of $N$ is reduced beyond what the accuracy of the path heuristics and assignment cost estimates can effectively predict, the assignments and tours included in the $N$-best ordered estimates may not reflect the best actual paths and assignments, jeopardizing the quality of the final assignment.

The pruning of poor tour paths and team assignments can only be as good as the path heuristics and team assignment estimates that are used
in the pruning. Effective path pruning comes when the tour heuristics properly represent the actual length of the path, and more importantly, when they properly represent the order of the tours from shortest to longest. The tour path heuristics used in pruning are calculated in nearly the same manner as the NILRTA* heuristics. The only difference is that the pruning heuristics include additional factors in calculating the path length heuristic that are intentionally left out of the NILRTA* path planning heuristics to satisfy the heuristic admissibility requirement of the A* algorithm. The additional factors are necessary because they prevent the heuristics from “breaking down” on smaller problems. In Figure 1.9, it can be seen that the heuristics begin to break down for the larger problems considered, resulting in the optimal assignment being found further down the ordered list of cost estimates.

Using the \( N \)-best assignments reduction method effectively reduces problem size while still producing improved team assignment results. Figure 1.10 shows that keeping only the \( N \)-best assignments reduces the number of individual tour paths needed for each individual vehicle to fully plan and calculate, in addition to reducing the number of assignments evaluated by the manager. The data shows that only a fraction of the possible individual vehicle tours are represented in the top \( N \) ordered assignments. Therefore, the \( N \)-best assignments assumption reduces the problem size and computational load for both the assignment manager and the individual UAV agents. By reducing the problem in this manner, improved assignments can be determined for near real-time applications.

The nature of the MVTV problem as outlined is similar to the multiple travelling salesman problem (or mTSP), with the added complication that each salesman is a dynamically constrained vehicle. The TSP and mTSP have been shown to be \( NP \)-complete problems [Motwani, 1992, Goldberg, 1993], and by extension, so too is the MVTV problem. The implication is that no algorithm other than a global search can guarantee the optimal or global solution. Maintaining a limited number of assignment estimates in the \( N \)-best assumption removes any guarantees that the solution will be optimal or even improved, but the accurate development and effective use of path length heuristics and assignment cost estimates has shown to reduce the problem size to a manageable level while still statistically improving the assignments that are returned. The motivation for such tradeoffs is the need for speed, which is discussed in Section 4.3.

At times the need for speed requires an even further reduction in the number of assignments kept (the value of \( N \)) than can be justified by the statistics shown in Figure 1.9. The ordering of assignment cost estimates
allows for this additional reduction. When the optimal assignment is not found, near-optimal assignments usually result. Figure 1.11 shows the average length of an assignment returned when compared to the length of the global solution. When keeping only 0.5 percent of the possible assignments for larger problems, the resulting path is only 10 percent longer than the global solution. This is still significantly better than the iterative assignments which are 78 percent longer, on average, than the overall assignment obtained with tour-planned paths. It is noteworthy that a smaller percentage of assignments is needed for large problems for effective solutions. This is significant because it demonstrates the feasibility of the proposed method for solving large problems in near real time.

4.3. Speed of Calculations

The size and complexity of the MVTV problem require certain trade-offs to be made between the optimality of the solution and the speed with which the result is returned for execution. The $N$-best assignment assumption increases the speed but also reduces the probability of obtaining the optimal assignment. The non-improving modification to the LRTA* planner has a similar result. By timing out of a non-improving
Figure 1.11. Assignment costs compared to the globally best tour assignment solution comparing the effectiveness of problem reduction methods and sizes.

tour path search, the planner increases the speed with which a path is planned but also decreases the probability that the path is truly optimal. The data shown in preceding sections demonstrate that these tradeoffs have not significantly compromised the ability to obtain better results through this method. The question that remains is whether the quality has been obtained with adequately low computational burden.

Assignments are obtained from the tour plan assignment method in sufficient time for execution in near real-time situations. The speed of the method is much slower than the Dubins paths/iterative assignment method used as a benchmark for comparison purposes, but it is not intended to be run as frequently. The assignment process only needs to be run a single time for an entire team assignment to be reached. By contrast, the iterative method runs every time the system state changes and a new subassignment needs to be made. Deciding whether or not the proposed assignment method is fast enough depends on a number of variables including

- the frequency of assignment calculations
- the amount of time in advance that agents know the target positions before assignment execution is required
- the quantity of previous calculations still applicable when the assignment needs to be recalculated
Cooperative Assignment through Tour Path Planning

- the level of confidence required in the solution

The speed of the algorithm depends on the computational capability of both the UAVs and the manager agent. The computation of the manager agent is primarily centered on three tasks: generating the complete and non-redundant set of vehicle tours and team assignments; calculating, evaluating and ordering of team cost estimates; and finally, the evaluation and ordering of actual assignment costs when vehicles report tour lengths and costs. For the manager, the amount of time required depends mostly on the number of total team assignments that are being kept and ordered (the value of \( N \)). Problem setup involves the first two steps mentioned. The assignment manager can entirely setup most problems, which would be considered small, in less than four seconds. Setup for global solutions (ordering all assignment cost estimates) for larger problems takes much longer, as can be seen in Table 1.3. In the table all targets are assumed to be visited three times each. In practice, the limit on the value of \( N \) has been set at 80,000 assignments that are explicitly kept and ordered so that setup can be fully executed on the order of seconds rather than minutes or hours.

**Table 1.3.** Setup times for problems of various sizes. All targets are assumed to be visited three distinct times.

<table>
<thead>
<tr>
<th>Number Vehicles</th>
<th>Number Targets</th>
<th># Assignments Kept &amp; Ordered</th>
<th>Avg Time to Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>1,000</td>
<td>0.4 sec</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>10,000</td>
<td>2.8 sec</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>8,000</td>
<td>2.1 sec</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>160,000</td>
<td>5 min</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1,500,625</td>
<td>2 hrs</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>75,000</td>
<td>6.6 sec</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>20,000</td>
<td>4.7 sec</td>
</tr>
</tbody>
</table>

The calculation of the individual tour path trajectories can be fully distributed to the several UAV agents. Complete NILRTA* paths involving multiple targets and tasks are calculated in 1.1 seconds\(^1\) on average. Actual times range between 0.2 and 1.8 seconds depending on the size of the world, the length of the path, the number of targets and the spread of their positions, and the number of tasks assigned in the tour. Problems solved in this work ranged from 16 to 512 tour paths per vehicle.

---

\(^1\)Computations were performed on a desktop computer with an AMD Athlon 2700 chip and 1024 MB RAM.
Global solutions require each vehicle to calculate all tours, but as Figure 1.10 shows, the individual UAVs are generally asked to plan only a fraction of the total possible tours when using the $N$-best assignments assumption.

5. Conclusions

The MVTV problem poses significant challenges for both path planning and task assignment. Path planning challenges include dynamical vehicle limitations and spatial coupling of targets and tasks. Task assignment is made more difficult by the need to prepare for both immediate needs and for future tasks. Path planning and task assignment are also coupled, leading to complications in determining effective path plans and assignments.

MVTV problems can be successfully addressed through the use of an improved tour planner that plans near-optimal paths through a sequence of multiple targets. Tour trajectory planning is accomplished through a Non-Improving LRTA* search. The NILRTA* search is effective at planning flyable paths for dynamically constrained vehicles. Through the search process, vehicles learn the best trajectory through a set of targets by taking advantage of the full sensor footprint to help overcome the spatial coupling of targets and individual tour segments.

Finally, improved assignments are made that specifically take advantage of tour-planned paths. When assignments are made using tour-planned paths the cooperative team can accomplish tasks in less time. Exponential growth in problem size can be controlled sufficiently through initial ordering of paths based on heuristics and team assignment estimates. Ordering by estimated cost leads to effective assignments, improved cooperation, and better use of team and individual resources. The resulting paths and assignments can be computed in near real time.

References


REFERENCES


