Location-Aware, Flexible Task Management for Collaborating Unmanned Autonomous Vehicles

Meng Wang, Yang Zhao and Alex Doboli
Department of Electrical and Computer Engineering
State University of New York at Stony Brook
Stony Brook, NY 11794-2350
adoboli@ece.sunysb.edu

Abstract

Unmanned Autonomous Vehicles (UAVs) are emerging as a breakthrough concept in technology. A main challenge related to UAV control is devising flexible strategies with predictable performance in hard-to-predict conditions. This paper proposes an approach to performance predictive collaborative control of UAVs operating in environments with fixed targets. The paper offers detailed experimental insight on the quality, scalability and computational complexity of the proposed method.

1 Introduction

Unmanned Autonomous Vehicles (UAVs) are emerging as a breakthrough concept for numerous applications in environment and infrastructure monitoring, defense, manufacturing, transportation, and so on [6, 7, 8]. The related applications involve a functionality (behavior) that must be achieved under strict constraints (e.g., deadlines, precision, safety, energy resources, etc.) while optimizing costs, like the value (utility) of the achieved objectives, the time required to meet the objectives, and the spent energy. In addition, UAVs must safely operate in hard-to-predict or even unknown environments and conditions, including moving obstacles and dynamically emerging threats [16, 13]. This poses interesting new challenges for the development of decision making (control) systems, which ought to offer optimized and reliable response in both predictable and hard-to-predict conditions.

UAVs sense the environment through a rich set of sensors and compute a response that is delivered through various actuators, including those used for moving. UAVs perform a large set of activities, including the computation of trajectories and identification of the control values for traveling along a trajectory, signal sampling and processing (including image processing), communication with other UAVs as well as specific activities, such as target detection and handling, and assessment of the results [1, 4, 5, 8].

In many instances, UAVs must operate collaboratively, so that complex activities can be tackled jointly by a group of UAVs [6, 7, 8, 12, 23]. The nature of collaboration is often decided dynamically at run time, depending on the context-specific situations. For example, an UAV might not be unable to meet the deadlines set for its tasks due to unforeseen overheads, such as the time required to avoid moving obstacles. In this case, the UAV might inquire neighboring UAVs whether they can collaborate for jointly performing the tasks. UAVs with more flexible deadlines might decide to satisfy the request, and participate to collaboration.

An important challenge relates to the need to offer predictable and reliable operation in hard-to-predict environments and situations. Traditionally, reactive control has been the *de facto* solution for situations that cannot be characterized off-line. Depending on conditions identified online, the controller selects the most suitable response from a set of predefined strategies. Each response strategy is characterized by specific outcomes and performance, like execution time, energy consumption, and so on. While certain "fixed-point" properties can be proven for reactive behavior (like stability and reachability) [12, 18, 19], properties, which depend on dynamic attributes (e.g., the frequency of being in a state), are harder to prove unless restrictive functioning conditions are assumed. Thus, important performance attributes, i.e. execution time and resource (energy) consumption, are hard to correctly estimate and guarantee for reactive control procedures. The alternative to reactive procedures are off-line static control methods [7, 15]. These methods work very well if the operation conditions and the environment are fully known, and hence the UAV behavior is deterministic. The performance of the control methods can be precisely estimated in this case. However, static methods have little or no flexibility in adapting to unknown situations. In conclusion, it is challenging to devise general, performance-predictable control strategies for efficient operation in dynamic conditions.

This paper proposes an approach to devising performance predictive methods for collaborative control of UAVs operating in environments with fixed targets. The control strategy allocates targets to UAVs and schedules in time the handling of the targets assigned to the same UAV, so that the associated deadlines are all met. The second optimization criterion is to maximize the flexibility of UAVs in participating to collaborative actions in which multiple UAVs jointly handle the same target. This goal is tackled by computing the slack time margins that can be used by an UAV in collaborative actions while still meeting the deadlines of its assigned tasks. Note that the approach does not select statically the UAVs, which participate in collaboration as this would limit their flexibility of operating in unknown situations. Instead, each UAV decides on-line whether it responds or not to a request for collaboration depending on its available slack time at that precise moment. If the slack time is less than the computed margin then the UAV can participate without violating the deadlines.

The proposed decision making approach assumes a two-level control hierarchy: the upper level strategy is a static procedure which decides the allocation and scheduling of the fixed targets to individual UAVs. The lower level strategy is reactive, and controls a UAV's responses to requests for collaborative activities. The reactive components decides based on inputs from the UAV's sensors as well as the slack time constraints allocated to the UAV through the static decision making process. This paper describes an Integer Linear Programming (ILP) based model for assigning and scheduling the fixed targets to UAVs and computing the slack time intervals used for collaborative actions. The ILP model was used the basis for developing an heuristic algorithm for task management. The experimental results for the heuristics are discussed in the paper.

The paper is organized in six sections. Section two summarizes the related work. Section three defines the addressed problem, and presents the modeling solution used for assigning and scheduling tasks to UAVs. Section four discusses the ILP modeling of the problem, and Section five offers experimental results. Finally, conclusions are put forth.

2 Related Work

This section summarizes existing work on decision making and control for autonomous systems.

Reynolds [14] proposes the concept of distributed control through aggregation of interacting actors. The purpose of control is to avoid actor collisions, velocity matching, and flock centering in computer animation. Brogan and Hodgins [3] present distributed control techniques for groups with significant dynamics. Similar to the method by Reynolds, there are only simple interactions between neighboring individuals. Interactions include communication, cooperation, and coordination strategies. Individuals have inertia, which introduces restrictions on the gradients for position and velocity. Therefore, the computed trajectories and velocity control rules must not only address collision avoidance and flocking but also prevent steep changes in direction and velocity.

Brock and Khatib [2] describe a method for distributed control through superposition of global and local requirements. Global requirements are static (e.g., the starting and ending points of a trajectory), and are modeled similar to virtual elastic strips that would span the two points. Any changes from the trajectory, i.e. to avoid an obstacle, introduces an elastic force that attempts to pull back the robot to the planned trajectory. Local requirements, like obstacle avoidance, are expressed through local interactions by repulsive forces between the obstacles and the moving robots. Then, the elastic and repulsive forces are used for velocity tuning and trajectory modification. In addition, the method relies on global information, e.g., robot priorities, and global trajectory re-planning to avoid any collisions. The method by Yamaguchi [23] uses attraction forces to targets and neighbors, and repulsive forces to obstacles for coordinated pattern formation. The author proves stability of the control rule, but does not suggest a general method for finding the local behavior rules for any global requirements.

Leonard and Fiorelli [9] suggest 2D distributed vehicle control based on artificial potentials and virtual leaders. Vehicles interact following potential fields following a logarithmic law. There is an attraction force to distant neighbors, so that group forming is encouraged, and there are repulsive forces and velocity matching with the close neighbors. In addition, virtual leaders define an additional local potential, which aids group formation. Finally, dissipative forces aim at matching a vehicles velocity to a desired velocity value. Vehicle control is guided by forces defined by the logarithmic laws of the four potential fields. The authors also indicate the mathematical rules that lead to certain formations, like equilateral triangle and hexagonal lattice.

Another artificial field based approach is described by Mamei, Zambonelli, and Leonardi [10]. Distributed control is based on computational fields (called Co-Fields), which act as an enabling global infrastructure between mobile agents. Co-fields are dynamic, and are influenced by moving actors. Actors move along the field waveforms, following either downhill, uphill, or equipotential lines. The goal of vehicle control is avoiding collisions, forming predefined patterns, and vehicle meeting. The authors indicate that finding the Co-Fields required for a certain goal is still an open research problem. The method presented in [16] combines nonlinear model predictive control with potential field concept for conflict-free trajectory planning. Other field based control methods are discussed in [20, 18].

Schouwenaars, Valenti, Feron, How, and Roche [15] discuss UAV trajectory generation for cooperative missions between manned and unmanned vehicles. A MILP (mixed integer linear programming) method is proposed for optimized real-time trajectory planning to avoid obstacles and threats in a partially known environment. A cooperative task scheduling method is presented in [6].

Rathbun *et al.* [13] propose path planning algorithms for UAV navigation in uncertain environments. The position of obstructions is known only with a limited accuracy. Trajectories are found using evolutionary algorithms (EA) that implement the following operators, (i) mutate and propagate, (ii) crossover, (iii) go to goal, and (iv) mutate and match. The obstacle intersection probability is defined for both moving and fixed obstacles. As optimization is carried out, more information is gathered about the position of the obstacles, which improves the accuracy of the probability functions. The cost function expresses the goal to reach the desired target, minimizing the amount of used fuel, and minimizing the probability of UAV collision with an obstacle. As the position of the obstacles becomes known, trajectories are re-planned such that they maximize the probability of success, and minimize the change in trajectory (as compared to the already existing trajectory). Other EA based approaches are presented in [5, 4].

Trajectory planning based on graph search methods using A* and D* methods are discussed in [11, 17].

Similar to the technique in [15], the proposed method uses ILP to find the task scheduling for UAVs. However, it considers a group of UAVs which can cooperate to meet the requirements of the application. The described algorithm finds the optimal path for handling the geographically-distributed tasks in addition to resource allocation and task scheduling. Producing flexible solutions is also a novelty of the proposed method as compared to similar work.

3 Problem Description and Modeling

This section defines the studied problem and presents the proposed modeling method.

Figure 1 summarizes the UAV behavior scenario. Multiple UAVs must cooperatively tackle targets located in a 3D environment. Each UAV moves along a non-linear trajectory at a variable speed using a trajectory computing algorithm similar to [22, 21]. UAVs are heterogeneous, and they might have different dynamic characteristics (e.g., speed and acceleration). The speed magnitude and speed gradients are bounded.

Fixed targets are positioned at known locations, and must be tackled before a predefined time limit. Otherwise, the entire mission is considered to be compromised. The group of UAVs must tackle all fixed targets before their time limit expires. The flexibility requirement states that the solution should maximize the chances of completing the mission in case UAVs experience unexpected conditions that delay their activities. Fixed obstacles are present in the 3D environment, and must be avoided by the moving UAVs.

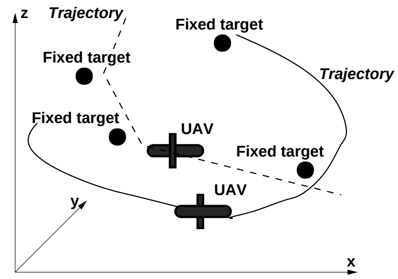

Figure 1. Cooperative operation of multiple UAVs

The tackling of targets comprises of the following sequence of activities: (i) flying to the target, (ii) detecting the target (e.g., through different sensors), (iii) handling the target (such as taking the picture of the target), and (iv) assessing the results of the activity [7, 15]. The three latter activities have known execution times, but the first step might take variable durations, depending on the position and flight characteristics of the UAV. Each UAV can handle multiple targets.

UAVs can collaboratively handle the same fixed target. For example, one UAV might detect and handle the target, and then transmit all the information to another UAV, which will do the assessment of the results. This scenario is useful considering that UAVs have different capabilities (e.g., achievable speed), and resources (such as amounts of fuel). A slower UAV positioned nearby can do the assessment, while the more powerful UAV moves towards tackling the next target. As a trade-off, the collaborative scenario involves communication overhead for information transmission between the UAVs, and also additional distances to be traveled (thus, higher fuel consumption) by the UAVs involved in collaboration.

In summary, the addressed problem is as follows. An algorithm must be developed for tackling N fixed targets by M UAVs with known but different characteristics. Each fixed target must be tackled before its predefined time limit expires. In addition, the flexibility of tackling new targets (known only at execution time) must be maximized.

3.1 Collaborative Approach

Figure 2 presents the decision making approach that we are proposing in this paper for controlling the behavior of

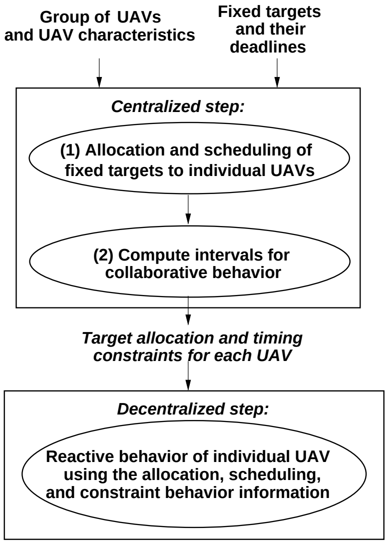

Figure 2. Decision making in collaborative UAV operation

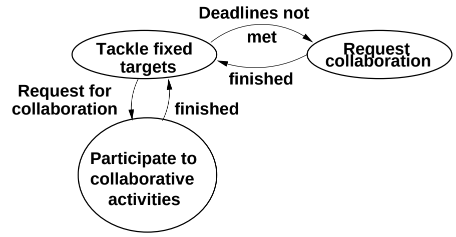

Figure 3. Decentralized controller for UAV operation

UAV groups. The approach represents a trade-off between centralized decision making, which is efficient and offers predictable performance (e.g., satisfaction of the predefined deadlines), and decentralized control, which is more scalable and flexible in tackling new situations. Centralized decision making is at the level of the entire UAV group, and decentralized control is at the level of each individual UAV.

The approach uses an off-line, centralized decision making step to compute the allocation of fixed targets to each UAV, and the scheduling in time of the activities related to the handling of a target. In addition, the centralized step also calculates the constraints that encompass the collaborative behavior of each UAV. During operation, each UAV decides dynamically (after a collaboration request has been formulated) whether it will participate to the collaboration, or not. The decision is made depending on its current status and geographical position so that the deadlines of its allocated targets are not violated. As collaboration requests are formulated dynamically and cannot be predicted off-line, the optimization goal is to maximize the chances of an UAV to participate to collaborations by computing the allocation and scheduling solution that maximizes the flexibility of an UAV to participate to collaborations.

Each UAV executes its own decentralized controller, which implements a reactive behavior expressed through a Finite State Machine (FSM). The controller decides the specific actions of an UAV, e.g., pursuing a fixed targets, or satisfying a request for collaboration. Figure 3 shows the structure of a simplified reactive controller. In normal operation mode, the UAV is tackling a fixed target following the allocation and scheduling decisions of the centralized step. If the controller detects that the deadlines fixed for the target currently being handled cannot be met, it formulates a request for collaboration. If the request is granted by another UAV then the UAV moves on to handling the next assigned target. If the request is not granted then the UAV might decide to continue with the current activity even though this results in violating the target's deadline, or leaving the task unfinished in order to move on to the next assigned target.

The focus of the paper is on the centralized decision making algorithm, including target allocation and scheduling, and computing of constraints related to the collaborative actions of the decentralized controller. The next subsection presents UAV behavior modeling for centralized decision making.

3.2 Problem Modeling

Figure 4 illustrates the task graph for tackling N fixed targets. Each target tackling activity is an independent thread of tasks consisting of separate tasks for flying, detection, handling, and assignment. Task dependencies express the required order of performing the tasks. As the N targets are known, the times for detection, handling, and assignment are fixed for a given UAV type. Please note that these times are different for UAVs with different characteristics. In contrast, the flight time is not known in advance because the time required for reaching a target depends on the computed UAV trajectory. Moreover, the UAV trajectory depends on the position of the fixed target previously tackled by the UAV, and therefore on the previous decisions on target allocation and scheduling.

Fixed targets. The task graph includes tasks that can represent a collaborative behavior between multiple UAVs in tackling the same fixed target. For example, after the target was detected by an UAV, the method allows that other UAVs handle and/or assess the target. These actions are represented by conditional blocks (the blocks labeled as "same UAV?" in the figure), which continue with the tasks for communication and flight, if a different UAV is involved. The communication time is fixed (as the amount of data to be transferred is given), but the flight time is variable as it

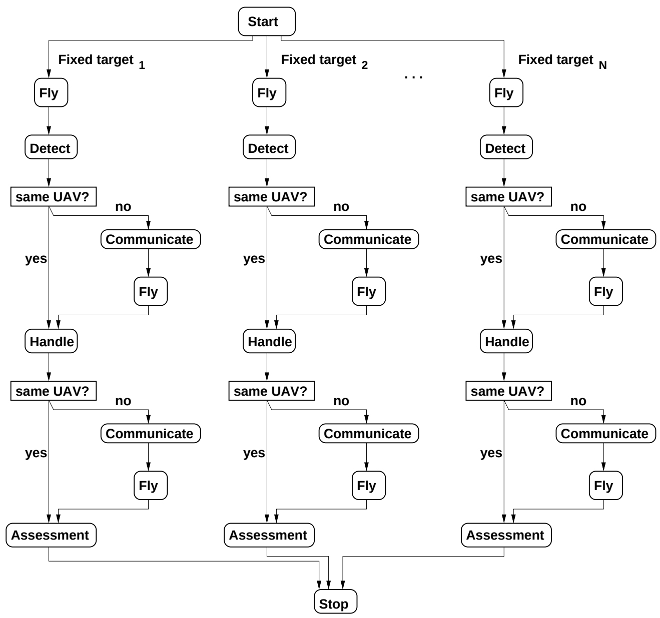

Figure 4. Task graphs for target tackling

varies with the position of the UAV entering the collaboration. The flight time includes the total time spent by a UAV for moving for a new activity as well as the time for accomplishing that activity. Since the nature of the collaborative activity is decided on-line, the actual flight time is not known during the step of off-line centralized decision making, and instead the methodology should maximize the overall capability of a UAV group for collaborative activity.

4 Proposed Algorithm

This section describes the centralized task assignment and scheduling problem as an Integer Linear Programming (ILP) problem. The model can then be solved using an existing ILP solver or an heuristic algorithm to obtain the centralized controller of an UAV group. Figure 4 is used to explain the ILP expressions. The following equations are used to build the ILP model:

1. Task start time

The start time of task i is larger than the end time of its preceding task j:

$$t_{i,start} \quad \geq \quad t_{j,end} \tag{1}$$

2. Task end time

$$t_{i,end} \quad = \quad t_{i,start} + x_{1,i}\, T_1 + x_{2,i}\, T_2 + ... + x_{M,i}\, T_M \tag{2}$$

The end time for executing task i (e.g., detection, handling, and assessment) is equal to the start time of the task plus the time T_i required for UAV$_i$ (i = 1, ..., M) to perform the task. Values T_i are constants for a set of UAVs. Variable x_i is one, if the the task is performed by UAV$_i$, otherwise it is zero.

3. Task allocation to UAVs

Each task pertaining to a fixed target must be allocated to exactly one UAV, which performs the task. For task k, this requirement is expressed as follows:

$$\sum_{i \in UAVs} x_{i,k} \quad = \quad 1 \tag{3}$$

4. Task scheduling to UAVs

Each UAV can handle multiple fixed targets, one target at a time. The set of ILP equations must include relationships that constraint the UAV to execute a single task at a time. For the tasks pertaining to the same fixed target, these constraints are implicitly introduced by the equations (1), which represent the sequencing constraints of the tasks.

For the tasks related to different fixed targets allocated to the same UAV, the constraint is that the UAV tackles a new target only after it finished tackling the current target. Allowing the UAV to intertwine the tackling of the two targets would result in unnecessary time overhead due to the extra distance the UAV must travel between the two fixed target. The overhead obviously affects the optimality of the scheduling result.

A 0/1 variable $z_{i,j}$ is defined for each pair of fixed targets i and j. If both targets are tackled by the same UAV, then the variable being one indicates that task i is tackled before task j, and after task j, if the variable is zero. This constraint is captured as follows:

$$\begin{aligned} T_{i,end} \leq T_{j,start}\, z_{i,j}\, & \textstyle\sum_{k \in UAVs} x_{k,i}\, x_{k,j} \\ & + T_\infty \, (2 - z_{i,j} - \sum_{k \in UAVs} x_{k,i}\, x_{k,j}) \end{aligned} \tag{4}$$

$$\begin{aligned} T_{j,end} \leq T_{i,start}\, (1 - z_{i,j})\, & \textstyle\sum_{k \in UAVs} x_{k,i}\, x_{k,j} \\ & + T_\infty \, (z_{i,j} + 1 - \sum_{k \in UAVs} x_{k,i}\, x_{k,j}) \end{aligned} \tag{5}$$

T_∞ is a very large value.

5. UAV flight time to fixed targets

The flight time T_{fly} to a fixed target depends on the UAV's previous position, which results from the fixed target allocation and scheduling. Target allocation and scheduling is computed by solving the ILP equation, and the allocation and scheduling variables are obviously unknown at the time of setting-up the ILP equations. The proposed solution is to introduce a 0/1 variable $w_{i,j}$ for each pair i and j of fixed targets to describe that the same UAV successively tackles the two targets (one immediately after the other). If

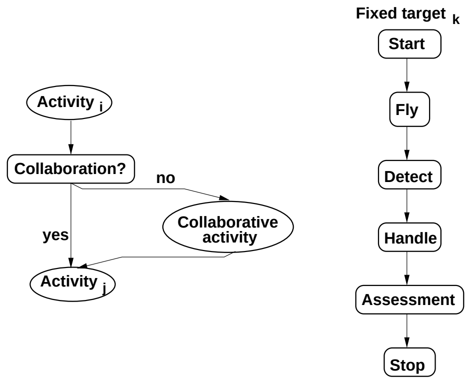

Figure 5. Modeling dynamic collaboration

the variable is one then target i is handled right before j. Otherwise, the variable is zero. In addition, the same UAV must tackle both tasks.

The flight time T_{fly} to a fixed target j is defined as follows:

$$T_{j,fly} = \sum_{\forall target_i} w_{i,j}\ Dist(target_i, target_j) \times \left(\sum_{\forall k \in UAVs} x_{i,k}\ x_{j,k} \right) \quad (6)$$

The next constraint expresses that each task i is tackled by one UAV after the UAV handles exactly one task (with the exception of the "dummy" start node):

$$\sum_{\forall target_j} w_{i,j} = 1 \quad (7)$$

6. UAV collaboration

In collaboration, the identity of the collaborating UAVs and the nature of the activities involved in collaboration is not known for centralized decision making, but instead is decided during UAV operation. The centralized decision making assigns and schedules tasks so that the flexibility of collaboration (if needed) is maximized.

As shown in Figure 5, a UAV might decide to collaborate after each of the activities related to a task, such as the fly, detect, and handle activities. The flexibility for collaboration depends on the time slack between the end of the current activity and the beginning of the next activity, and the deadline of the target handling for which the collaborative action is requested. The more overlapping exists between the slack time and the deadline the more flexibility exists in collaborating to meet the deadline. If there is no slack time or no overlapping with the deadline (e.g., the deadline is before the starting of the slack time, or after the end of the slack time) then there is no possibility of the UAV to participate in handling the target.

Figure 6 is used to explain the ILP equations for collaboration. For each target k, we define $SetC_k$ as the set

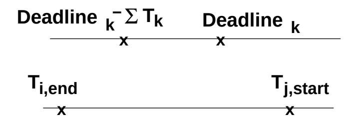

Figure 6. Flexibility for collaboration

of targets for which the assigned UAVs are candidates for collaboration. $SetC_k$ can be identified statically based on the geographical proximity of the targets (this information is known), or can be decided dynamically (at run time) depending on the current slack time of an UAV, hence its flexibility to fly to more distant targets without violating the deadlines of its assigned targets. In this paper, we have assumed that $SetC_k$ is static.

The flexibility for participating to a collaborative handling of target k between activities i and j (scheduled in this order) is proportional to the following value:

$$Fl_{i,j,k} = [Activity_{i,end}, Activity_{j,start}] \cap [deadline_k - \sum T_k, deadline_k] \quad (8)$$

Variables $Activity_{i,end}$ and $Activity_{j,start}$ are the end time of Activity i and the start time of Activity j. $[Activity_{i,end}, Activity_{j,start}]$ represents the time interval defined by the two moments, $deadline_k$ is the deadline set for target k, and $\sum T_k$ is the time required to perform all activities related to target k, e.g., detect, handle, and assess.

Figure 6 illustrates the definition of the flexibility constraint for UAVs collaborating on the handling of target k. Targets i and j are allocated to the UAV. The condition for collaboration is defined by the following equations:

$$T_{i,end} \leq deadline_k - \sum T_k \quad (9)$$

$$deadline_k \leq T_{j,start} \quad (10)$$

The equation for the flexibility for UAV m is then:

$$Fl_{i,j,k} = x_{m,i} x_{m,j} (1 - x_{m,k}) (deadline_k - T_{i,end})(T_{j,start} - deadline_k) \quad (11)$$

The total flexibility of an UAV to participate to collaborative activities is as follows:

$$Fl_{i,j} = \sum_{k \in SetC_k} Fl_{i,j,k} \quad (12)$$

The overall cost function includes a term to maximize the flexibility of UAVs participating to collaborative activities.

7. Cost function

The cost function is a weighted sum that express the goals of (i) minimizing the cumulated penalties for exceeding the predefined deadlines for the static targets, (ii) minimizing the total distance traveled by the UAVs, and (ii) maximizing the flexibility of the solution:

Ex.	# nodes	Total time	Best at (#)	Total #	Exec. time (sec.)
SN 1	11	64	23,244	43,244	49
SN 2	14	89	13,664	33,664	37
SN 3	20	164	26,874	46,874	84
SN 4	38	337	9,954	29,954	205
SN 5	50	649	100	20,100	151

Table 1. Optimization for minimum total time

$$Cost = \alpha \times \sum_{\forall targets_i} (T_{i,end} - T_{i,deadline}) + \\ \beta \times \sum_{\forall UAV_k} distance_k - \gamma \times \sum_{\forall\ targets\ i,j} Fl_{i,j}$$

5 Experimental Results

This section presents the experimental results for the proposed algorithms. An heuristic algorithm was developed for solving the ILP model presented in Section IV. The algorithm is based on Simulated Annealing. It minimizes the cost function while satisfying all the constraints of the ILP model. Using an heuristic algorithm instead of an ILP solver offers two important advantages: it scales better than solvers for large ILP problems, and it does not have convergence problems, which is important for the reliability of the method. The algorithm was implemented in C language and run on a SUN Sparc workstation.

The experimental set-up varied the number of the targets to be tackled, the number and characteristics of the UAVs, and the geographical position of the targets. Three different cost functions were optimized: (i) minimize the total execution time needed for tackling all targets, (ii) minimize the total distance traveled by the UAVs, and (iii) maximize the flexibility of the solution for collaboration between the UAVs. In addition to the quality of the solutions, experiments observed the computational characteristics of the heuristic algorithms, such as the execution time, the iteration at which the best solution was found, and the total number of iterations performed by Simulated Annealing. The scalability of the algorithm with the number of targets was also observed.

Table I presents the characteristics of the different experiments as well as the results obtained for minimizing the total time required for tackling all targets. This experiment was used as a reference for comparing the optimization results for minimizing the total distance traveled by UAVs and maximizing the flexibility, respectively. The second column indicates the number of nodes in the task graphs for target tackling (similar to Figure 4). The third column presents the minimum execution time for tackling all targets as found by the algorithm. Column four shows the iteration number at which the best solution was found. Column five indicates the total number of iterations performed by Simulated Annealing, and Column six presents the execution time of Simulated Annealing. As expected for an heuristic algorithm, the convergence does not increase with the problem size as the total number of iterations depends mainly on the stochastic dynamics of Simulated Annealing. The execution time increases with the problem size, however it remains reasonably large even for the larger examples. This indicates that the algorithm scales fairly well with the size of the problem. For the two smaller examples, we manually verified that the found results are optimal.

Ex.	Min.dist.		Min.time		Improv. (%)	
	Total distance	Total time	Total distance	Total time	Total distance	Total time
SN 1	56	81	64	70	12.5	13.5
SN 2	89	104	73	91	19.2	12.5
SN 3	200	256	280	164	28.5	35.9
SN 4	586	395	619	337	5.3	14.6
SN 5	601	356	650	349	7.5	1.9

Table 2. Optimization for minimum total distance

Ex.	Max. flex.	Init. flex.	Improv. (%)	Total time	Total dist.
SN 1	78	66	15	71	101
SN 2	112	16	85	99	155
SN 3	148	30	79	197	368
SN 4	272	76	72	408	873
SN 5	436	274	37	540	943

Table 3. Optimization for maximum flexibility

Table II presents the optimization results for minimizing the total length traveled by UAVs. Columns two and three indicate the total distance and the total time resulting for this cost function. For comparison purposes, Columns four and five show the total distance and total time found for the cost function minimizing the total time (also shown in Table I). Finally, Columns six and seven indicate the relative improvements in terms of total distance and total time between the two optimization requirements. Column six shows that the optimized paths can be 5.3% to 28.5% shorter than the total paths for solutions optimized for time. However, the paid penalty is in longer total time, which can be larger by values between 1.9% to 35.9%.

Table III offers results for resource allocation and scheduling optimized for flexibility. Column two presents the maximum flexibility. Column three gives the flexibility produced by a simple list scheduling algorithm. Column four shows the relative improvement. For comparison purposes, Columns five and six indicate the total time and total distance of the solutions optimized for flexibility. The experiments show significant improvement in the flexibility, between 15% and 85%. However, increased flexibility results at the penalty of longer times and traveled distances as compared to the previous two cost functions.

Figure 7 illustrates the nature of the solutions found for example $SN\ 1$ and two UAVs. Similar results were obtained also for the larger examples. Time minimizations always distribute targets to UAVs such that there is an equal loading of the two UAVs. The loading includes the execu-

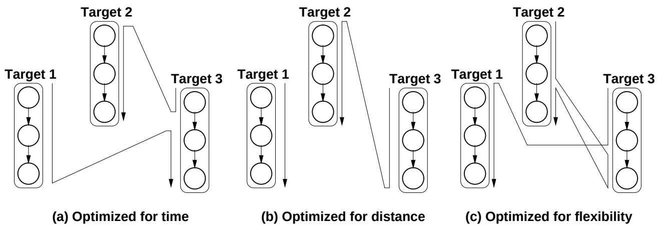

Figure 7. Results for different optimization criteria

tion times of the tasks as well as the distance traveled by the UAVs. Distance optimizations tends to assign clusters of neighboring targets to UAVs even though this might result in unequal loading of the UAVs. Finally, flexibility optimization encourages a scheduling such that the two UAVs perform as much as possible their assigned task in parallel.

6 Conclusions

Unmanned Autonomous Vehicles (UAVs) are emerging as a breakthrough concept in technology. A main challenge related to UAV control is devising flexible strategies with predictable performance in hard-to-predict conditions. This paper proposes an approach to performance predictive collaborative control of UAVs operating in environments with fixed targets.

Experimental results show that the proposed algorithm scales fairly well for large problems, has a reasonably long execution time, and can significantly improve the quality of the produced solutions, such as up to 28.5% reductions of the total path traveled by UAVs and up to 85% improvement in the flexibility of the solution.

References

[1] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, Sept/Oct. 1989.

[2] O. Brock and O. Khatib. *Real-Time Obstacle Avoidance and Motion Coordination in a Multi-Robot Workcell*. Proc. of IEEE International Symposium on Assembly and Task Planning, 1999, pp. 274–279.

[3] D. Brogan and J. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, Kluwer(4):135–153, 1997.

[4] B. Capozzi and J. Vagners. *Evolving (Semi) Autonomous Vehicles*. Proc. of AIAA Guidance, Navigation and Control Conference, 2001.

[5] D. Fogel and L. Fogel. *Optimal Routing of Multiple Autonomous Underwater Vehicles through Evolutionary Programming*. Proc. of Symposium on Autonomous Underwater Vehicle Technology, 1990, pp.44–47.

[6] A. Gil, K. Passino, and S. Ganapathy. Cooperative task scheduling for networked uninhabited air vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 2007.

[7] J. How, E. King, and Y. Kuwata. *Flight Demonstrations of Cooperative Control for UAV Teams*. AIAA "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, 2004.

[8] I. Kaminer and O. Yakimenko. *Cooperative control of small UAVs for Naval Applications*. 43rd IEEE Conference on Eecision and Control, 2004.

[9] N. Leonard and E. Fiorelli. *Virtual Leaders, Artificial Potentials and Coordinated Control of Groups*. Proc. of IEEE Conference on Decision and Control, 2001, pp. 2968–2973.

[10] M. Mamei, F. Zambonelli, and L. Leonardi. *Distributed Motion Coordination with Co-Fields: a Case Study in Urban Traffic Management*. Proc. International Symposium on Autonomous Decentralized Systems, 2003.

[11] J. Mitchell and D. Keirsey. *Planning Strategic Paths through Variable Terrain Data*. Proc. of SPIE Conference on Applications of Artificial Intelligence, 1984, Vol. 4, pp. 172-179.

[12] B. Moore and K. Passino. Decentralized redistribution for cooperative patrol. *International Journal on Nonlinear and Robust Control*, 2007.

[13] D. Rathbun and B. Capozzi. *Evolutionary Approaches to Path Planning Through Uncertain Environments*. Proc. of American Institute of Aeronautics and Astronautics, 2002.

[14] C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–43, 1987.

[15] T. Schouwenaars, M. Valenti, E. Feron, and J. How. Linear programming and language processing for human/unmanned-aerial-vehicle team missions. *Journal of Guidance, Control, and Dynamics*, 29(2):303–313, March-April 2006.

[16] D. Shim, H. Chung, H. Kim, and S. Sastry. *Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles*. Proc. of AIAA GN & C Conference, 2005.

[17] A. Stentz. *Optimal and Efficient Path Planning for Partially-Known Environemnts*. Proc. of International Conference on Robotics and Automation, 1994, Vol. 4, pp. 3310-3317.

[18] H. Tanner, A. Jadbabaie, and G. Pappas. *Stable Flocking of Mobile Agents, Part I: Fixed Topology*. Proc. of IEEE Conference on Decision and Control, 2003, pp. 2010-2015.

[19] H. Tanner, A. Jadbabaie, and G. Pappas. *Stable Flocking of Mobile Agents, Part I: Dynamic Topology*. Proc. of IEEE Conference on Decision and Control, 2003, pp. 2016-2021.

[20] H. Van Dyke Parunak, S. Brueckner, and J. Sauter. *Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles*. AAMAS, 2002.

[21] O. Yakimenko. Direct method for rapid prototyping of near-optimal aircraft trajectories. *AIAA Journal of Guidance, Control, and Dynamics*, 23(5):865–875, 2000.

[22] O. Yakimenko and V. Dobrokhodov. *Airplane trajectory control at the stage of rendezvous with maveuvering boject algorithms syntheses*. IEEE, 1998.

[23] H. Yamaguchi. A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations. *Robotics and Autonomous Systems*, Elsevier:257–282, 2003.