Evolution-Based Deliberative Planning for Cooperating Unmanned Ground Vehicles in a Dynamic Environment

Talib Hussain, David Montana, and Gordon Vidaver

Department of Distributed Systems and Logistics, BBN Technologies
Cambridge MA, USA, 02138
{thussain, dmontana, gvidaver}@bbn.com

Abstract. Many challenges remain in the development of tactical planning systems that will enable automated, cooperative replanning of routes and mission assignments for multiple unmanned ground vehicles (UGVs) under changing environmental and tactical conditions. We have developed such a planning system that uses an evolutionary algorithm to assign waypoints and mission goals to multiple UGVs so that they jointly achieve a set of mission goals. Our evolutionary system applies domain-specific genetic operators, termed tactical advocates because they capture specific tactical behaviors, to make targeted improvements to plans. The plans are evaluated using a set of tactical critics that together comprise a multiobjective fitness function. Each critic evaluates a plan against criteria such as avoiding an enemy or meeting mission goals. Experimental results show that this approach produces high-quality plans with the potential for real-time dynamic replanning.

1 Introduction

Recent advances in technologies for the control of unmanned ground vehicles (UGVs) have demonstrated the ability to perform local path navigation while traversing unknown, off-road terrain. Moreover, these technologies permit simple longer-range path planning, such as navigation between human-specified waypoints. However, the challenge remains to develop technologies for automated generation of plans that result in the achievement of higher-level mission goals (such as reconnaissance, surveillance, and target acquisition) despite changing environmental conditions, evolving mission requirements, and the need to coordinate multiple UGVs [1].

In response to an environment and a set of mission requirements that are dynamically changing, the planning system must perform replanning of both the *reactive* (local) and *deliberative* (global) varieties. Examples of reactive replanning are when a UGV avoids an obstacle or turns to run away from an enemy. An example of deliberative replanning is when a UGV discovers a previously unknown enemy and modifies its entire path to circumvent the enemy and remain hidden en route to its next mission goal. Another, more complex, example of deliberative replanning is when a UGV, after discovering an enemy and realizing that it can no longer reach its next

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1017–1029, 2004.

mission goal in time, trades goals with another UGV that has a clear path to the first UGV's mission goal. While there has been some previous work done on deliberative planning for robots and UGVs, progress has been slow, with much more practical work in the reactive planning area.

Our approach is to view the entire deliberative planning problem as an optimization problem to determine an operation plan for multiple UGVs that achieves multiple mission goals while satisfying multiple tactical criteria as best as possible based upon the most recent environmental and tactical situation knowledge available. An *operation plan* is defined as a set of paths, one for each UGV, in which each path is a sequence of navigation waypoints. A *mission goal* is defined as a geographical location or area that must be visited, along with some measure of the time at which that area should be visited. A UGV may be assigned zero or more mission goals. A *tactical criterion* is defined as a property of an operation plan that is desirable in the context of the current state of the environment, such as enemy avoidance, hazard avoidance, stealth or rapid achievement of mission goals.

This deliberative UGV planning problem shares some important characteristics with a classic optimization problem, the vehicle routing problem with time windows (VRPTW) [2]. In both problems, multiple vehicles need to move in such a way as to arrive at particular locations during particular time windows. However, the UGV planning problem has some critical extra complications. One is that the paths between locations are not well defined, and the planning algorithm must find a good path over some combination of roads and off-road terrain. A second complication is that there are a greater number of criteria to consider in determining a good plan.

We use an evolutionary algorithm to search for a good solution. For such a complex optimization problem, an evolutionary algorithm is a good approach. In addition to their ability to search efficiently through large and complex spaces, evolutionary algorithms offer the advantage of being easily tailored to a particular domain for improved performance. We take advantage of this with our use of tactical advocates and tactical critics. The advocates are domain-specific mutations that modify a plan based on knowledge about good tactics. The critics compute the different evaluation metrics corresponding to different criteria of what constitutes a good plan. The structure of the software and algorithm makes it easy to add new advocates and critics and hence to incorporate domain knowledge. We discuss in detail this evolutionary algorithm approach, which we refer to as Advocates and Critics for Tactical Behaviors (ACTB), in Section 3.

To validate our approach, we have developed test scenarios in which multiple UGVs cooperate to solve complementary and competing mission goals while minimizing mission completion time as well as minimizing risk to mission success. One such scenario incorporates the actual terrain that the Army uses as a testbed for some of its UGVs. As we discuss in Section 4, the system has demonstrated that constantly improved plans can be quickly generated, both before and during plan execution, in response to changes in the tactical situation.

2 Background

The deliberative planning problem we are investigating seems like it should be amenable to a variety of well-studied techniques. However, we now argue that these techniques do not actually apply.

One set of approaches that do not apply is the traditional Artificial Intelligence (AI) planning algorithms. Classical planning [3], hierarchical-task-network planning [4], and case-based planning [5] use symbolic planning based on logic and reasoning. However, this problem is essentially numeric and hence not suited to reasoning about goals and subgoals. The higher-level strategic planning problem, how to decide what the mission goals are, is potentially well matched to AI planning techniques, but we are interested in the tactical planning problem where the mission goals are already known.

A second set of techniques that largely do not apply is those for coordinated robot planning. Many have collision avoidance during path planning as a primary concern [6]. In our problem, there is so much space compared to the number of vehicles that the low-probability case of a potential collision can be handled by reactive planning, and we place our deliberative planning emphasis on how to share the work rather than avoid collisions. Other multi-robot planning algorithms are concerned with formations and moving in unison rather than dividing the workload [7]. Mataric does investigate a variety of ways of coordinating robot behavior by dividing the work, e.g. [8]. However, this workload decomposition is generally reactive rather than deliberative, losing the benefits of planning ahead for multiple goals. Furthermore, path planning is treated as a separate problem, thus not considering issues such as an enemy between a UGV/robot and a nearby goal point when assigning goals.

Perhaps the work closest to what we are doing is that by Carnegie-Mellon University (CMU) on control of UGVs. The core of the CMU control system is the Distributed Architecture for Mobile Navigation (DAMN) [9]. Among other features, DAMN provides a sophisticated reactive control component. DAMN contains behaviors, each of which represent some higher-level navigation goals, such as 'road following', 'seeking the next navigation goal', 'obstacle avoidance', 'avoid hazards'. Each behavior provides a vote on the next direction to take, and a command arbiter decides upon the best direction, which is then taken by the UGV. While most of the behaviors are reactive, there is one behavioral input from a deliberative planner called the global navigator [10]. The global navigator is capable of determining a full path to a goal position using a D* (dynamic A*) search algorithm. However, this approach still does not incorporate as many criteria and as much information at the deliberative planning level as we believe are necessary to determine mission assignments and paths that are not fooled by local gradients.

As we mentioned in Section 1, the problem we are solving is to first order a combination of the vehicle routing problem and robotic path planning. Genetic algorithms have been used for each of these tasks in the past. An example of a genetic algorithm for path planning is [11]; an example of a genetic algorithm for vehicle routing is [12]. The novelty of this problem is jointly solving the two problems, plus being able to adapt the solution dynamically to a changing environment.

3 Technical Description

3.1 System Design

The ACTB system addresses the need to perform continual deliberative planning within a dynamic environment in which UGVs move and knowledge regarding the environment and tactical situation may change. We have developed a simulation-based system in which a deliberative planning process explicitly interacts with a simulated world environment in a continual cycle, as illustrated in Figure 1. The deliberative planning process uses the ACTB genetic algorithm to evolve multiple notional operation plans for a fixed number of generations. After the genetic run, the best plan is then adopted as the current execution plan. In the simulated world environment, the execution plan is communicated to the UGVs, which use a simple (non-reactive) execution model to visit their waypoints. s execution proceeds, simulated world events, such as the discovery of a new enemy location, may occur. These events trigger the deliberative planning process to evolve a new plan that incorporates the new tactical situation. Additionally, at regular intervals, the execution process may be suspended and the deliberative process executed to explore further improvements to the current operational plan. The population of the genetic algorithm is persistent across runs.

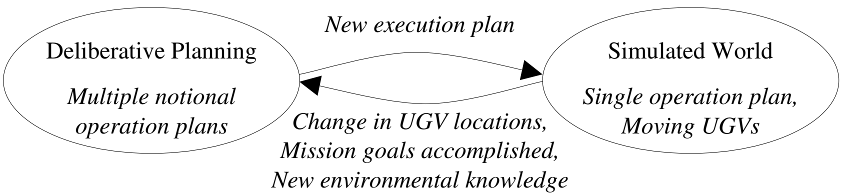


Fig. 1. Interaction between deliberative planning process and simulated world

ACTB is programmed in Java 1.4 and the simulation environment uses the OpenMap™ geographical system [13] to represent terrain information, provide basic functionality for making geographic inquiries, and provide a graphical interface.

3.2 ACTB Genetic Algorithm Design

The ACTB genetic algorithm is based upon the fundamental notion that significant improvements to a plan may be made through a succession of small, goal-directed changes. These goal-directed changes are made using domain-specific genetic operators termed *tactical advocates*. A tactical advocate promotes the use of a specific tactical behavior during deliberative planning, where a *tactical behavior* is defined as an action performed by a UGV that may generally satisfy one or more tactical criteria. For example, a tactical behavior may be to follow a road, as opposed to travelling cross-country. Such a behavior is tactical in that it may lead to an improvement in the speed with which the UGV accomplishes its tasks, or the rapidity with which it moves away from a known enemy. In addition to the tactical advocates, traditional domain-generic operators are also used to augment the search capabilities of the system and maintain diversity.

The ACTB genetic algorithm accommodates the multiobjective nature of the deliberative planning problem through the use of multiple, distinct evaluation components to determine fitness, thereby following an established approach for solving multiobjective optimization problems [14]. Specifically, a tactical critic represents a domain-specific evaluation component that computes a single term in a fitness function. Each tactical critic evaluates how well a given operation plan satisfies a tactical criterion. For instance, a critic for safety may evaluate a plan to determine how much danger the UGVs are placed in due to traveling too close to a known enemy. The outputs of multiple critics are combined using a weighted sum to form a single fitness value. In the military context, the weights associated with the critics reflect the tactical priorities of the operation.

The ACTB genetic algorithm accommodates the constraint-based nature of the deliberative planning problem by allowing ostensibly "illegal" individuals into the population (i.e., those that violate constraints) and using fitness values to reflect the magnitude of the violations. This is an example of an established approach for handling constraints [15]. Specifically, when a tactical critic evaluates a given operation plan against a tactical criterion, it assigns a penalty if the plan violates that criterion. For example, a critic to evaluate whether the path is traversable will accept a path that crosses water (an untraversable terrain), but assign a high penalty. To enable a relative judgement amongst "illegal" plans, critics will typically assign a penalty that is proportional to the degree of the violation. For example, the amount of distance "traveled" in water will determine the magnitude of the penalty.

An important property of the tactical critics is that they exploit the most recently available environmental knowledge. As such, the fitness of an individual plan in the population may vary whenever the environment state varies. In the simulated environment, the genetic algorithm may be run many times, each time for a small number of generations. The population is persistent across runs, but may require re-evaluation at the beginning of a run if the environment state has changed.

3.3 Genetic Representation

Given n UGVs, a genome is defined as a set of n chromosomes, where each chromosome defines the path for one of the UGVs as a variable length sequence of geographical locations, or *waypoints*. For the purposes of evaluating the fitness of a genome, every successive pair of waypoints is assumed to be connected with a straight line. Each chromosome therefore defines a piece-wise linear directed path. An important aspect of the genetic representation is that the first waypoint in each path represents the *next* waypoint of the corresponding UGV. The first segment in a path is inferred to be the straight line between the UGV's current location and the first waypoint in the path.

In order to enable effective genetic manipulations, the representation has three types of waypoints, each representing a different conceptual aspect of a path. A *mission-point* is a waypoint that attempts to satisfy a given mission goal at a specific location, and the sequence of mission-points determines the order in which the UGV accomplishes its assigned missions. A *route-point* is a waypoint that marks a specific location on the map, and a sequence of route-points is used to determine the general route followed by the vehicle between two mission-points. A *travel-point* is a waypoint that marks a specific location on the map, and a sequence of travel-points is used to specify a detailed route followed between two route-points. A key feature of travel points is that they are not available for selection as points of genetic manipulation. Rather, they are used to incorporate specific path segments between two consecutive mission or route-points. The relative benefits of these segments may then be evaluated through genetic search. In the current system, route-points are used by the road-following advocate (see below) to represent complex road segments.

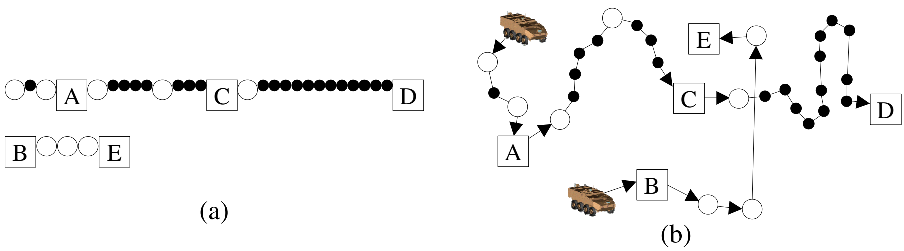


Fig. 2. (a) Genome with two chromosomes as sequences of three types of waypoints, and (b) inferred path for each UGV from its current location

Figure 2a illustrates a sample genome that has two chromosomes. Each mission-point is flagged with the name of the mission at that location (e.g., A, B, etc.) and is represented as a square. Each route-point is represented as an open circle, and each travel-point is represented as a solid circle. Figure 2b illustrates the geographical locations of the waypoints and the inferred directed path for each UGV among its waypoints, starting at the UGV's current location. Note that UGV paths may cross.

The three types of waypoints are used to enable genetic manipulations targeted to different levels of planning. For example, manipulation of mission-points and their order performs the task of scheduling missions, while manipulation of route-points serves performs the task of route planning. In the context of a UGV platform, no distinction is required between the three types of waypoints.

3.4 Tactical Advocates

Three tactical advocates are used in the current ACTB system. The *Mission-allocation* advocate exploits knowledge of the mission goals and their requirements to allocate mission goals to the UGVs. It operates in two modes. The first mode is selected if there is an outstanding mission goal that has not been assigned to any UGV within a given genome. In this mode, the advocate assigns an outstanding mission goal to a randomly selected UGV by inserting a mission-point into the path next to the existing route-point or mission-point that is closest to the mission. The order of the new mission is determined by the existing information in the UGV's path. The second mode is used when all missions have been assigned within a plan. The advocate randomly removes a sequence of mission-points (one or more) and all intermediate waypoints from a path, and inserts that sequence before or after a randomly chosen mission-point on a randomly chosen path in the plan. Thus, the missions may be inserted within the same UGV's path, thereby performing an effective re-ordering of the mission goals, or in the path of a different UGV, thereby performing a switch of mission goals between UGVs. In this offspring, the new ordering of missions is randomly determined.

The *Avoid-untraversable* advocate exploits terrain knowledge and a model of the movement capabilities of the UGVs to determine routes that do not have waypoints in untraversable terrain. For instance, rivers and lakes may be untraversable. The advocate identifies all waypoints over all chromosomes in a genome that lie in untraversable terrain. It randomly selects one of these waypoints and moves it to a location on traversable terrain. The new location is selected by searching on an arc towards the "traversable-predecessor" of the selected untraversable-waypoint. Any intervening waypoints are eliminated (since they necessarily would have been untraversable).

The *Road-following* advocate exploits knowledge of the road network to determine a path segment between any given pair of mission or route points that makes maximal use of roads. All roads are represented symbolically within Openmap. The road-following advocate randomly chooses two mission or route points on a randomly chosen chromosome. Using deterministic routines that query the Openmap road representation, the advocate first determines the closest road point to each selected waypoint and then obtains the shortest road path between those road points. This road path is represented as a sequence of travel points with route-points at the ends. The new sequence replaces the path between the original selection points.

The use of travel-points rather than route-points to determine road following is important. A segment of road may be highly curved or irregular, and as such require a large number of points in order to specify that segment in a piece-wise linear manner. If the travel-points were included as possible points of selection by the other advocates and genetic operators, the process of selecting waypoints for adaptation would be overwhelmed by the large number of travel-points. For example, the nudge-waypoint operator (see below) would spend the majority of its time moving road points, and thus be highly ineffective at optimizing the route between mission goals.

A first glance, the road lookups of the road following advocate seem to serve a purpose similar to the shortest path lookups of existing planning techniques. However, the road lookup is limited to identifying only small road segments, and has no impact upon the exploration of cross-country paths by other genetic operators. Rather than simply looking up a shortest path between two mission goals, which may be tactically poor, the ACTB genetic algorithm uses multiple genetic operators to determine routes, and creates routes by making a number of small changes at random locations in the chromosome. This enables the GA to explore a wide variety of routes between the two mission goals and adapt that route according to multiple tactical criteria. For example, some segments may result in a poor fitness according to one tactical critic, and eventual removal or modification of those segments through genetic operators and advocates may produce an improved path according to that critic.

3.5 Genetic Operators

Several traditional mutation and crossover genetic operators are used within ACTB to maintain enough variability in the population so that the tactical advocates continue to make novel plans rather than continually rehashing old ones. Three mutation operators are used, each of which performs a mutation on a randomly chosen chromosome within the genome. Only mission-points and route-points are manipulated. The *insert-waypoint* mutation operator will randomly select a waypoint on the chromosome and insert a single waypoint before or after that point. The geographic location of new waypoint will be a small random distance in a random direction from the line connecting its neighbors. The *remove-section* mutation operator will randomly select two waypoints in the chromosome and remove them and all waypoints between them. The *nudge-waypoint* mutation operator will randomly select a waypoint from the chromosome and modify its geographical location slightly in a random direction. Two crossover operators are used. The *path-crossover* operator is applied to a single genome parent, and performs variable-length one-point crossover between two randomly chosen chromosomes within the genome. The *plan-crossover* operator is applied to two genome parents, and performs variable-length one-point crossover between a randomly chosen chromosome in one parent and a randomly chosen chromosome in the other.

3.6 Tactical Critics

Five tactical critics are used in the current ACTB system, and all return evaluations that are greater than or equal to 0, where lower numbers indicate better plans.

The *Traversability* critic exploits terrain knowledge and a model of the movement capabilities of the UGVs to identify all portions of the path that cross untraversable terrain. It returns a penalty proportional to the distance traveled on untraversable terrain over all chromosomes (i.e., we allow a path to cross untraversable terrain but penalize accordingly).

The *Safety* critic exploits knowledge of the known enemy locations and a model of enemy capabilities to evaluate whether a given plan puts one or more UGVs in danger by placing them too close to a known enemy. It returns a penalty proportional to the distance traveled within danger range of any enemies over all chromosomes (i.e., we allow a path to cross dangerously close to enemies, but penalize accordingly).

The *Stealth* critic exploits knowledge of the known enemy locations and line-of-sight computations to evaluate whether a given plan puts one or more UGVs at risk by placing them in the line-of-sight of a known enemy. Line-of-sight is computed using an Openmap routine and a model of the surveillance capabilities of the enemy. The advocate returns a penalty proportional to the distance traveled within surveillance range of the enemy (i.e., we allow a path to cross within sight of enemies, but penalize accordingly).

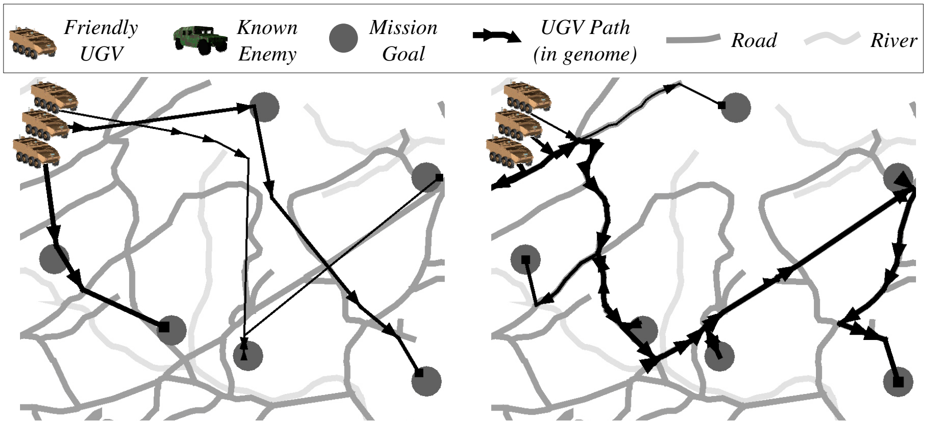
The *Mission-success*, *Total-duration* and *Max-duration* critics use a deterministic greedy algorithm to interpret how a given chromosome would be executed by a UGV. The algorithm assumes a model of UGV movement speed over different terrain types and evaluates the travel time between successive mission-points based upon the distances and terrain traversed over the (piece-wise linear) path between them. Each UGV is assumed to travel as fast as possible between mission points, and then wait as little as needed (if early) to meet the time window requirement (i.e., the greedy choice). Thus, no special representation of time windows is required in the genome. The mission-success critic evaluates how well a given plan comes to accomplishing all mission goals, and returns a penalty proportional to the number of failed goals and degree of failure. The total-duration critic evaluates how long each UGV takes to execute its chromosome, and returns the sum of the durations of all chromosomes. The max-duration critic evaluates how long each UGV takes to execute its chromosome, and returns the longest duration over all chromosomes.

4 Experimental Results

The ACTB system was tested under four conditions to demonstrate the effectiveness of the tactical critics for multiobjective optimization and examine the search capabilities of ACTB when using tactical advocates in conjunction with traditional genetic operators over using traditional genetic operators alone. The experiment examined the basic tactical route planning capabilities of the ACTB system. Time scheduling aspects of the problem were minimized by making the mission time windows very wide. However, path duration was still an important factor (i.e., do all the missions as soon as possible).

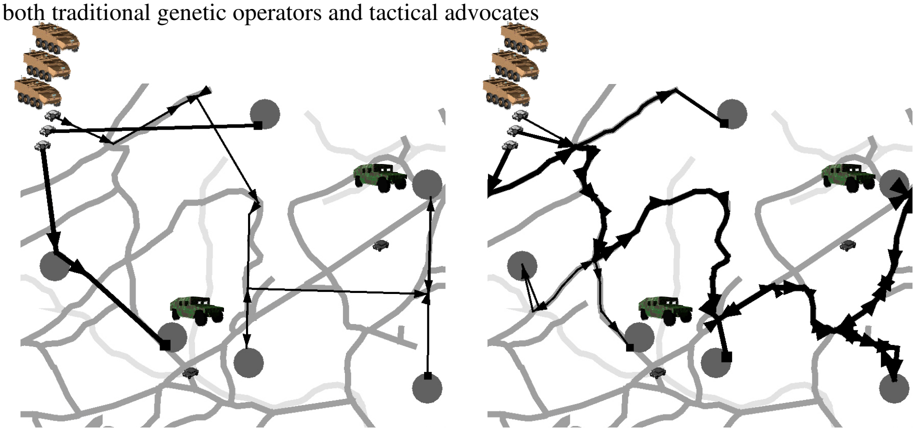
In all conditions, a steady-state genetic algorithm was run using a fixed population size of 50, fitness-proportional selection was used, and offspring competed with all members of the population. Most advocates and genetic operators were applied with the same likelihood of selection (i.e., 1.0). To encourage the system to explore complex paths, *insert-waypoint* was applied with twice the likelihood of the above (i.e., 2.0), and *remove-section* with half the likelihood (i.e., 0.5). Critic weights were selected to assign a very high penalty to untraversable portions of the routes and to missed missions, a moderate penalty to exposure to the enemy (i.e., completing the mission is more important than avoiding the enemy), and a small penalty to path duration; the penalty for *maximum-duration* was weighted twice as strongly as *total-duration* to encourage a more equal distribution of mission goals among UGVs.

Final plans developed in the four experimental conditions are illustrated in Figures 3 and 4. In the first pair of conditions (Figure 3a and 3b), the system examined basic routing in a simple situation with no known enemies. In the second pair of conditions (Figure 4a and 4b), the system examined tactical routing in a situation with two known enemies. In the figures, three friendly UGVs are located in the top left and each UGV's path is indicated by a different line thickness; off-road terrain is shown in white, and all intersections of road with water are bridges. Travel on-road is assumed to be roughly 10 times faster than travel off-road. These figures show that in all conditions, the system was able to evolve plans that were traversable, met all mission goals and distributed mission goals among all three UGVs. In both enemy conditions, the evolved paths avoided the known enemies.



(a) (b)

Fig. 3. Evolved paths with no known enemies using (a) traditional genetic operators and (b) both traditional genetic operators and tactical advocates



(a) (b)

Fig. 4. Evolved paths with two known enemies using (a) traditional genetic operators and (b) both traditional operators and tactical advocates

Figure 5 illustrates the fitness of the best individual each generation in each experimental condition. Each condition was run for 30000 generations to ensure convergence, but all conditions had converged by 7000 generations. The range of fitness values was very large due to high penalty values assigned by the tactical critics, and the results are graphed on a logarithmic scale to emphasize the changes over evolution. A fitness value under 1000 indicates no penalty except for duration of path. The ACTB system clearly demonstrated better plans when using advocates, as illustrated by the better (i.e., lower) fitness values overall and by the rapid achievement of a plan with no major penalties around 1000 generations as opposed to over 4000 generation for the traditional conditions. The final plan generated in the advocate conditions completed all missions in roughly half the time of the plans generated in the traditional conditions. As illustrated in Figures 3b and 4b, this improvement is clearly due to improved road following.

We have also tested the system in a dynamic simulation mode, as described earlier, in which new enemies may be detected as the UGVs are executing an operation plan. The ACTB system has demonstrated the capability for rapid and effective replanning in response to these changes in the tactical situation, as illustrated in Figure 6. Figure 6a illustrates a plan under execution immediately before the discovery of the enemy. Figure 6b illustrates the re-planning activity initiated upon the discovery (after a few generations). Figure 6c illustrates the new plan generated after 300 generations and passed to the UGVs for execution. Note that after replanning, all UGV paths avoid the area surrounding the enemy.

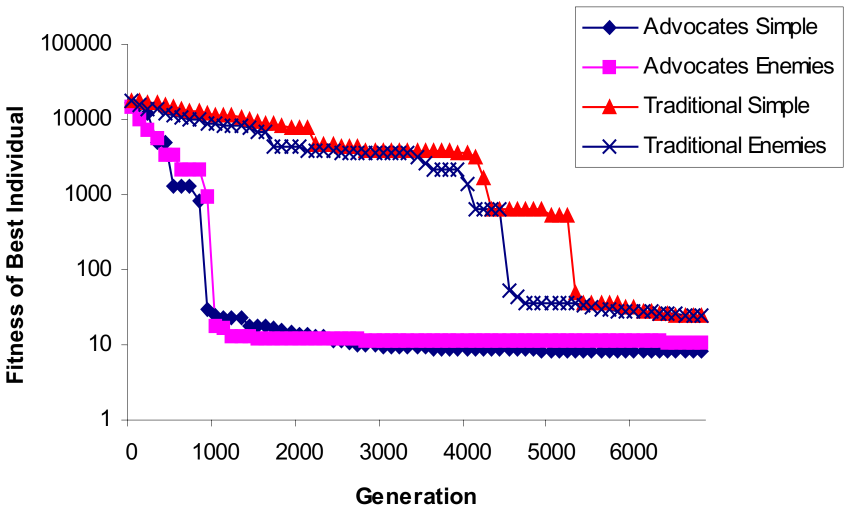


Fig. 5. Fitness of best individual by generation for all four experimental conditions

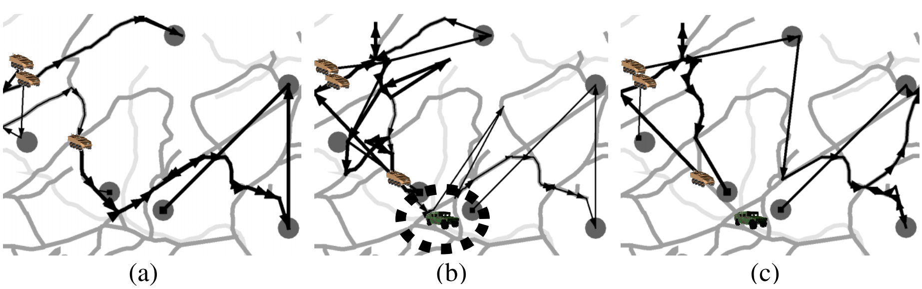


Fig. 6. Sequence of three plans illustrating (a) execution plan before, (b) new plan immediately upon and (c) new plan shortly after discovery of a new enemy

5 Conclusion

We have implemented and evaluated an approach to deliberative planning for coordinating UGVs. This approach is based on representing the planning problem as an optimization problem and using a genetic algorithm to search for a good solution. Multiple evaluation components, called tactical critics, enable the evolution of plans satisfying multiple tactical constraints. Domain-specific operators, called tactical advocates, greatly speed the search process yielding rapid plan turnaround. By continually searching for improvements to the plan, we ensure that the plan will adapt to changes in the tactical situation. We have provided preliminary evidence that the ACTB system maintains good plans in response to such changes. We are exploring the development of advocates and critics for additional tactical behaviors and mechanisms for distributing the evolutionary algorithm to make ACTB amenable to implementation within UGV platforms directly.

Acknowledgements. We would like to acknowledge the efforts of Stephen Milligan, Richard Lazarus and Disk Estrada in the development of the ideas presented in this paper, and the efforts of Aaron Iba, Brian Krisler and Sarah Siracuse in the development of the simulation environment used to test our approach.

References

1. National Research Council Staff: Technology Development for Army Unmanned Ground Vehicles. National Academies Press, Washington, D.C. (2002)
2. Solomon, M.: Algorithms for the vehicle routing problem with time windows. Transportation Science **29** (1995) 156-166
3. Fikes, R., Nilsson, N.: STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence **2** (1971) 189-208
4. Nau, D., Cao, Y., Lotem, A., Muñoz-Avila, H.: SHOP: Simple hierarchical ordered planner. In: International Joint Conference on Artificial Intelligence (1999) 968-973
5. Spalazzi, L.: A survey on case-based planning. Art. Intelligence Review **16** (2001) 3-36
6. Svestka, P., Overmars, M.: Coordinated path planning for multiple robots. Robotics and Autonomous Systems **23** (1998) 125-152
7. Balch, T., Arkin, R.: Behavior-based formation control for multi-robot teams. IEEE Transactions on Robotics and Automation **14** (1998) 926-939
8. Gerkey, B., Mataric, M.: Sold!: Auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation **18** (2002) 758-786
9. Rosenblatt, J.: The distributed architecture for mobile navigation. Journal of Experimental and Theoretical Artificial Intelligence **9** (1997) 339-360
10. Brumitt, B, Stentz, A.: Dynamic mission planning for multiple mobile robots. In: IEEE International Conference on Robotics and Automation (1996) 2396-2401
11. Ashiru, I., Czarnecki, C., Routen, T.: Characteristics of a genetic based approach to path planning for mobile robots. Network and Computer Applications **19** (1996) 149-169

12. Baker, B., Ayechew, M.: A genetic algorithm for the vehicle routing problem. Computers and Operations Research **30** (2003) 787-800
13. BBN Technologies. Openmap™: Open System Mapping Technology. openmap.bbn.com
14. Coello, C.: A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems **1** (1999) 269-308
15. Coello, C.: A survey of constraint handling techniques used with evolutionary algorithms. Technical Report Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada (1999)