

ETL-0571

AD-A239 718



Cooperative Autonomous Agents Testbed First Annual Report

Charles Dolan
David Payton
Karel Zikan

Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

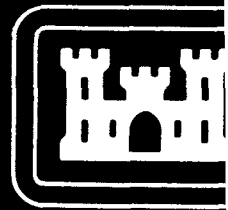
August 1990

Approved for public release; distribution is unlimited.

Prepared for:
U.S. Army Corps of Engineers
Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546

01 0 00 041

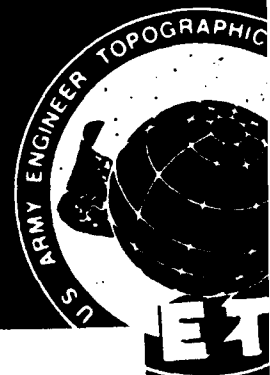
91-09010



E

T

L



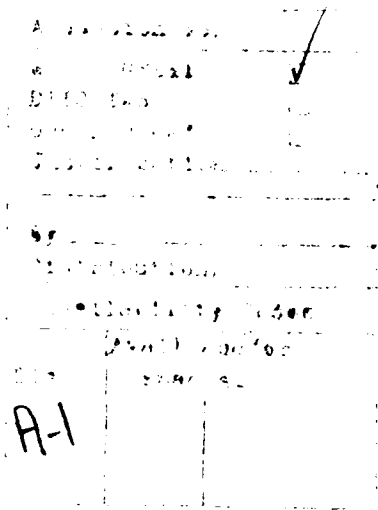
ETL

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official
Department of the Army position unless so designated by other
authorized documents.

The citation in this report of trade names of commercially available
products does not constitute official endorsement or approval of the
use of such products.

CONTENTS



TITLE	PAGE
FIGURES	iv
PREFACE	v
1. BACKGROUND AND SCOPE	1
1.1 Guide to the Report	2
2. RESUPPLY AS A MULTI-AGENT PLANNING PROBLEM	2
2.1 Evaluation of Planning Algorithms	5
2.2 Internalized Plans for Multi-agent Problems	9
3. USING THE MATHEMATICS OF ECONOMICS	11
3.1 Setting Prices	12
3.1.1 The Energy Function	13
3.1.2 Choosing New Flows	14
3.2 Mechanisms of Price Modification	15
3.2.1 Relationships and Contrasts between the Centralized and Distributed Planning	16
3.2.2 Quadratic-Payoff Economy	18
3.2.3 Cobb-Douglas Economy	19
3.3 Decomposing Graphs into Cycles	20
4. KNOWLEDGE ACQUISITION: EVALUATING THE APPROPRIATENESS OF OUR MODEL	21
5. USING SIMNET AS A TESTBED	23
6. CONCLUSIONS	24
7. REFERENCES	24

FIGURES

TITLE	PAGE
1 Resupply routes are constrained both by varying traversability of terrain and by crossing point for natural barriers.	3
2 Natural constraints on resupply routes create opportunities for predatory behavior against an enemy's supply operations.	4
3 Natural movement constraints and opportunities for preying on enemy supply routes allow us to represent a realistic multi-agent problem domain as a graph structure.	4
4 The ownership of a node by a team is shown by the different style of shading. Nodes in contention are shaded with both styles. Circular shaded nodes indicate sources of supplies, square nodes indicate conflicts.	6
5 By shifting more of its supply activity to a node, one team can take a contended node for itself. Node ownership is a continuous variable as shown by the change in the amount of cross hatching of the nodes in the highlighted area of the graph.	6
6 When one team changes the amount of supplies flowing to a node, the other team is forced to react, or yield the contended node.	7
7 In a slight change of notation, fully shaded circular nodes indicate supply sources, doughnut shaded circular nodes indicate nodes through which a team's supplies may safely pass or be cached, and square nodes indicate conflicts. Note that in this figure, conflict occurs along an arc connecting two square nodes. This notation allows us to represent "cutting supply lines" in the graph notation.	8
8 By sending extra supplies to the top gray node, the gray team can move forward along that conflict arc.	8
9 By representing a plan as a gradient field, instead of as a set of way points, an autonomous vehicle has more flexibility in using cues from the environment to direct its actions. Without this flexibility, time can be wasted achieving meaningless sub-goals.	10
10 Example of a random point selection.	15
11 Example of a decomposition of a feasible flow into cycles. Numbers on the arcs indicate the amount of flow on the arc. In the second graph, dashed lines indicate circular flows of constant amount.	21
12 A cycle decomposed into simple cycles.	21

PREFACE

This report describes work performed under Contract DACA76-88-C-0008 by Hughes Research Laboratories, Malibu, California 90265 for the U.S. Army Engineer Topographic Laboratories (ETL), Fort Belvoir, Virginia 22060-5546 under the sponsorship of the Advanced Concepts and Technology (ACT) Committee, U.S. Army LABCOM, Adelphi, Maryland 20738-1145. The Contracting Officer's Technical Representatives at ETL were Rosalene M. Holecheck and Kevin Mullane. The ACT point of contact is Dr. R. Gonano.

Yearly Technical Progress Report 1990
Contract No. DACA76-89-C-0002
Cooperative Autonomous Agents Testbed

Charles Dolan
David Payton
Karel Zikan

1 Background and Scope

The goal of this project is to study methods by which multiple autonomous agents can be made to interact effectively. The primary emphasis is on mobile, automated and autonomous agents and on problem domains of interest to the Army. To this end, it is our intent to study problem domains which incorporate both cooperation and competition between agents. By creating a testbed in which teams of autonomous agents can interact, we will be able to study how individual agents must cooperate within a team in order to compete, as a group, with an adversary team. Cooperative scenarios allow the examination of the relationships between centralized control and distributed decision-making.

Competition is the primary method used in evaluating various planning architectures and implementations. By having teams of agents compete against one another, the strengths and weaknesses of different planning methods can be compared and analyzed. The performance of the agents can be incrementally improved in an evolutionary process in which the best features are extracted from existing agents and synthesized into new agents.

There are four primary issues to be addressed in creating a useful testbed for autonomous agents.

1. Find a problem domain where human agents currently work cooperatively but where there is future potential for autonomous agents.
2. Develop a formalism for expressing planning problems in the problem domain.
3. Create algorithms that generate plans using the formalism.

4. Design agents that can use the plans to operate in a changing environment.

Because of the limited scope of this effort, when choosing a problem domain, we pay close attention to the cost of simulating autonomous agents in the world.

This report covers a five (5) month technical effort. Although a contract was awarded for a 24 month technical effort, by agreement with the contract monitor, work was stopped after five (5) months to allow Hughes to develop some general purpose simulation tools under Hughes IR&D. Those simulation tools have been completed and work under contract is continuing. The work using the new simulation tools is not covered by the period of this report.

1.1 Guide to the report

In Section 2 we define a graph formalism for looking at multi-agent resupply problems. This formalism allows us to use a great deal of mathematics from economics and optimization, as described in Section 3. In addition, we have found that graph-based resupply problems can be better analyzed by decomposing graphs into simple circular flows as described in Section 4. In section 5 we cover our discussions with Army personnel about the applicability of our graph formalism to resupply problems today. Our future work is described in Section 6.

2 Resupply as a multi-agent planning problem

We began by defining a multi-agent problem domain which provides a suitable environment for analysis while also presenting a level of complexity which will help maintain the relevance of this work to real-world problems. The domain we developed combines a basic resupply problem with aspects of strategic predation. The types of competition between agents and the degree of dynamics in the environment can be varied for experimentation with different types of scenarios. The primary advantage of this is that we can vary the complexity of the problem without making radical changes to the environment. Thus, planning skills developed in simpler environments can remain applicable as the complexity of the problem domain is increased.

Figure 1 shows a sample problem in resupply. Note that natural terrain obstacles, roads, and safe conduct corridors impose restrictions on the movement of agents. Therefore, one view of the resupply problem is to maximize the flow from consumers to producers along a network or graph.

Our view of the resupply domain is fairly general and is intended to reflect problems ranging all the way from the delivery of parts on a factory floor to the delivery of weapons to battle fronts. Simply put, autonomous agents in our environment will attempt to carry objects from producers to consumers by the most efficient means possible. Their travel will be restricted to limited

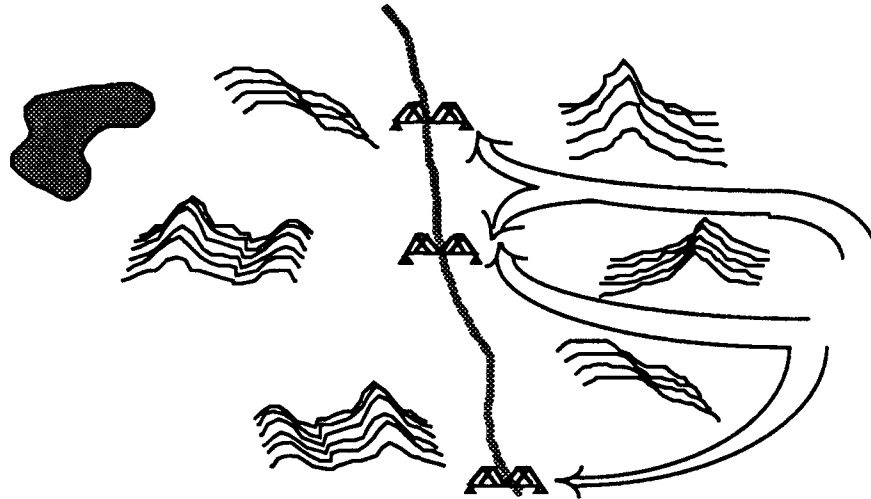


Figure 1: Resupply routes are constrained both by varying traversability of terrain and by crossing point for natural barriers.

pathways, so agents must coordinate their activities so as to avoid interfering with one another.

This same domain may be applied to the study of strategic predation problems. To do so, we will allow two teams to compete within the same environment, and provide each team with the ability to cut off the supply lines of their opponents. Figure 2 shows predation as part of the resupply problem. By establishing a superior level of supplies at a battle front site, a team may win battles and capture territory, thereby interfering with enemy supply routes. By surrounding a battle site, a team can reduce the opponent's ability to supply the battle with needed resources, thereby eventually winning that battle and gaining new territory. The effort to surround, however, will entail the creation of new battle fronts, which will put the aggressor at greater risk of loss. In this way, the environment combines the need for efficient supply strategies with the need for strategic planning for predatory operations.

Our observation of the strong movements constraints imposed on resupply vehicles leads us to the conclusion that this domain can be effectively represented as a graph. Figure 3 shows a graph superimposed on the terrain features from Figures 1 and 2. By casting the problem domain in the language of graphs, we can now bring to bear a number of mathematical tools. But is this formalism too limited for studying cooperation among teams of agents?

We feel that the graph view of resupply has a very intuitive mapping onto many problems. Of course, for problems (or portions of problems) for which

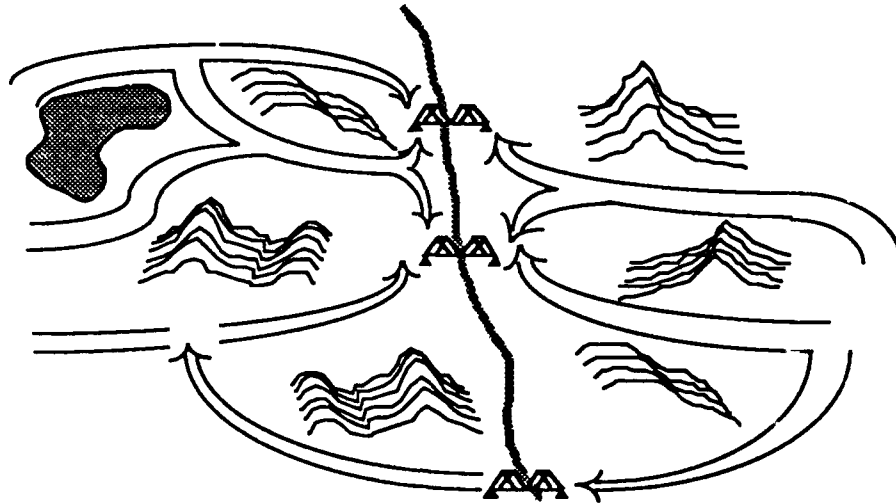


Figure 2: Natural constraints on resupply routes create opportunities for predatory behavior against an enemy's supply operations.

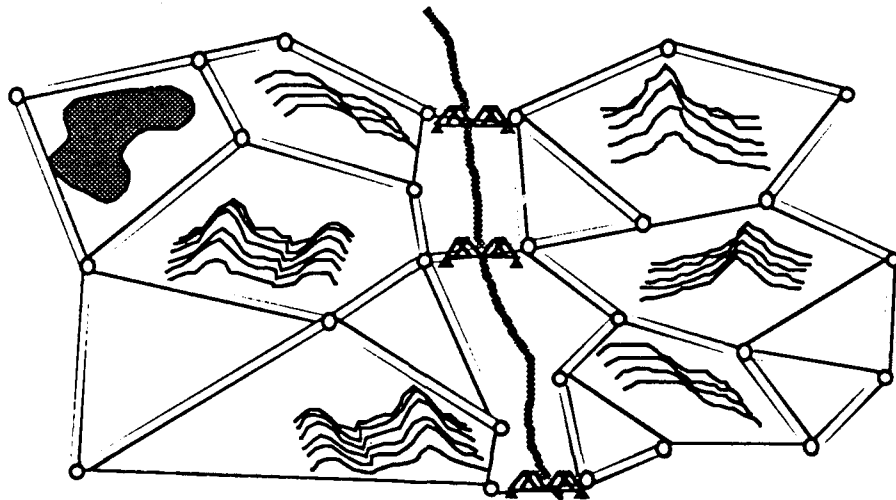


Figure 3: Natural movement constraints and opportunities for preying on enemy supply routes allow us to represent a realistic multi-agent problem domain as a graph structure.

graphs are not a good match, we can place the nodes on the terrain with arbitrarily fine granularity connected on a quad or hex grid. While this is not recommended for large problems, saturating a small area with nodes can be used to represent small open areas of relatively unrestricted mobility.

2.1 Evaluation of Planning Algorithms

The levels of task complexity achievable in our domain are important because they define the types of activities that will be required of our autonomous agents. In domain we have three levels of complexity.

1. Preassigned sources and destinations for each resupply vehicle.
2. Dynamic assignment of sources and destinations.
3. Exchange of commodities *en route* under agent direction.

At the simplest level, we have pre-assigned sources and destinations between which agents must travel in order to perform their resupply task. Since agents are modeled as trucks that can transport supplies, even the simplest environment presents the problem of optimizing the simultaneous and opposing flows of empty and full trucks. The next level of complexity is achieved by allowing agents to select their own sources and destinations for pick-up and delivery of supplies. This gives agents the ability to adapt to changing rates of production and consumption. Another type of complexity arises when we allow agents to exchange payload at intermediate locations along their route. This can be used by a team to improve the efficiency of a delivery route.

In order to create competition between teams, we can consider reflecting a network about the delivery sites, and install a team in each half-network to feed the same delivery sites. We can now think of these delivery sites as battles. By comparing the relative contribution of supplies from each team at a given battle site, we can obtain a measure for victory or loss at that site. Since both teams will attempt to win as many battles as possible, we have a suitable environment for competitive evaluation.

We can see an example competitive scenario in Figures 4-6. We see in Figure 4 that the cross hatched team is devoting more resources to the two conflict nodes in the large rectangle. In the resupply domain, this corresponds to sending more fuel and ammunition to a particular part of a battle. In Figure 5 we can see the graphical depiction of the cross hatched team gain control of two nodes. Figure 6 shows the gray team's response, sending more supply to the heavily contested nodes. We can encourage agents who anticipate the enemy by making the cost of *reaching* parity at a node larger than the cost of *maintaining* parity.

A further complication to this scenario is to allow battles to move as they are won or lost. In this way, a team can encroach upon the territory of the

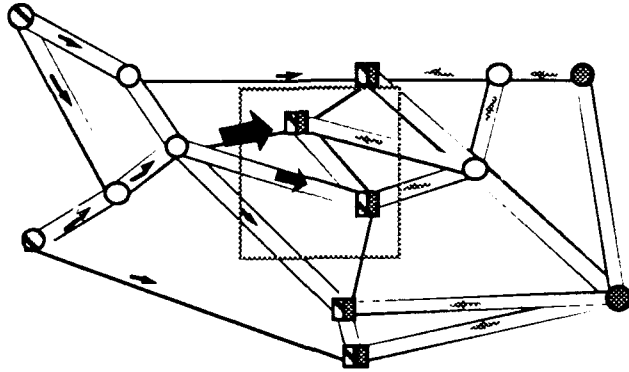


Figure 4: The ownership of a node by a team is shown by the different style of shading. Nodes in contention are shaded with both styles. Circular shaded nodes indicate sources of supplies, square nodes indicate conflicts.

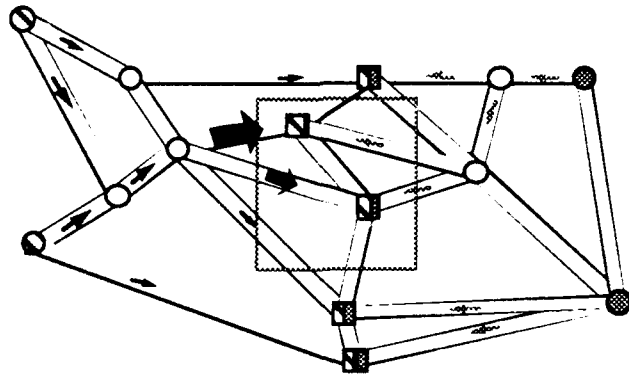


Figure 5: By shifting more of its supply activity to a node, one team can take a contended node for itself. Node ownership is a continuous variable as shown by the change in the amount of cross hatching of the nodes in the highlighted area of the graph.

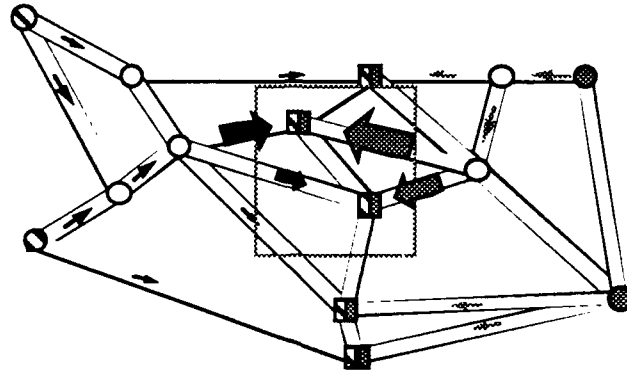


Figure 6: When one team changes the amount of supplies flowing to a node, the other team is forced to react, or yield the contended node,

other by winning a sufficient number of battles in a given area. Furthermore, by pushing a battle into the opponent's territory, existing supply lines may be cut off so that other ongoing battles are more easily won. In this way, the resupply problem leads naturally to strategic predation.

Figures 7 and 8 demonstrate how strategic predation works in a graph based problem. In Figure 7 we see that conflict occurs along arcs connecting nodes held by different teams. This change from our convention in Figures 4-6 allows two important phenomena to be modeled, (1) concentrating force and (2) cutting supply lines. In Figure 8 we see that the grey team has advanced one node at the top of the graph. This effectively cuts-off one of the cross hatched nodes and also concentrates force on it. This small scenario shows how competition among teams makes resupply on a graph a dynamic competitive environment.

One important objective of this contract is to be able to quantitatively evaluate different planning methodologies. The scenarios described above provide us with the ability to compare two different teams of autonomous agents either by ranking them according to a set of absolute performance measures or by noting their success in direct competition. Both the resupply and the strategic predation domains are amenable to this kind of evaluation. In the resupply domain, it should be possible to measure the overall efficiency of a team in meeting a fixed demand. By comparing the efficiencies of different teams, we will be able to determine which team performs best in different situations. The resupply domain also affords us with various opportunities to subject teams to direct competition. In these scenarios, the actions of one team will necessarily be influenced by the actions of the other. The team that can enact the best strategies while responding most readily to change will inevitably be the most successful. As we move into strategic predation scenarios, notions of evaluation through direct competition become even more critical. In these tests, a team's

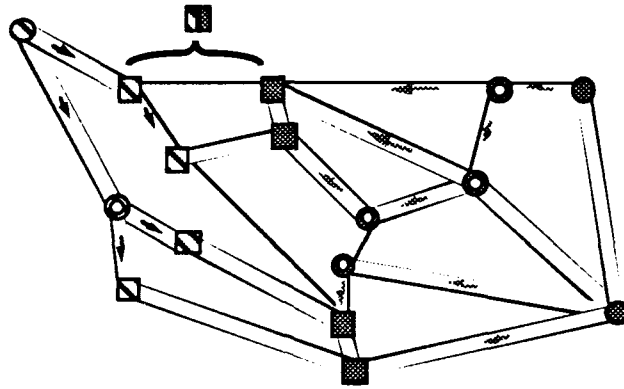


Figure 7: In a slight change of notation, fully shaded circular nodes indicate supply sources, doughnut shaded circular nodes indicate nodes through which a teams supplies may safely pass or be cached and square nodes indicate conflicts. Note that in this figure, conflict occurs along an arc connecting two square nodes. This notation allows us to represent “cutting supply lines” in the graph notation.

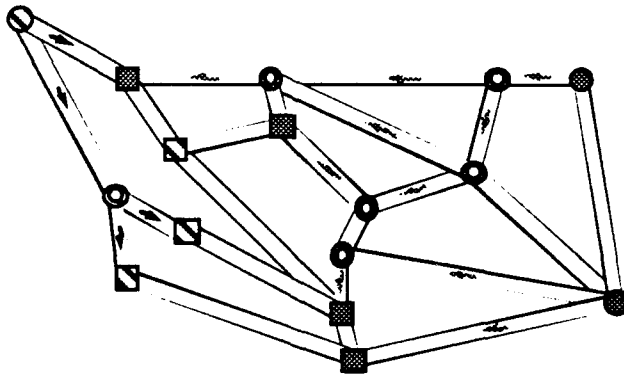


Figure 8: By sending extra supplies to the top gray node, the gray team can move forward along that conflict arc.

environment will be almost entirely defined through interaction with the opposing team so there would be no way to evaluate the team's performance in isolation.

2.2 Internalized Plans for Multi-agent problems

As more planning researchers work on mobile robots, it is becoming clear that standard sub-goal based plans are not appropriate for agents in dynamic, noisy environments. The top portion of Figure 9 shows a standard, sub-goal based plan for an autonomous land vehicle (ALV) from (Daily *et al.* 1988). The ALV, shown in the lower right hand corner of the figure, is controlled from a radio tower and so the plan avoids an RF shadow caused by a rock outcrop. When this plan was actually executed, the vehicle accidently did go into the supposed RF shadow while avoiding a collision with the rock outcrop. The RF shadow turned out to be no problem, but the vehicle still headed towards the subgoal at the top of Figure 9. Even though a much shorter route would have been to go directly to the main goal!

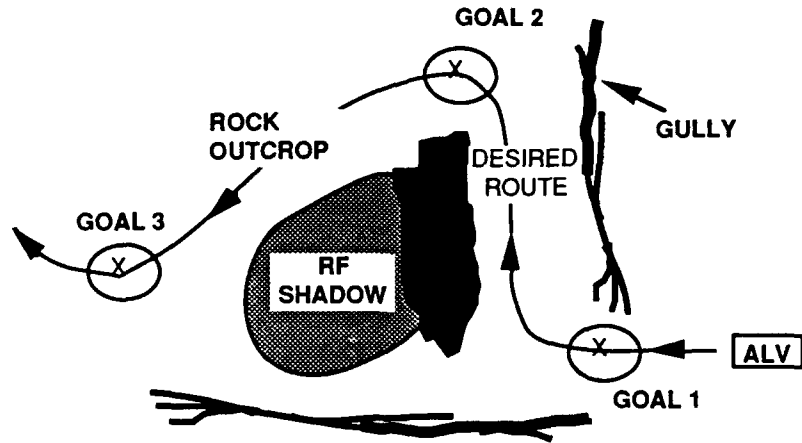
This problem can be solved by a technique called internalized plans (Payton 1988). The bottom half of Figure 9 shows an internalized plan for the ALV. The internalized plan is a gradient field showing the best direction to move from any point on the map. The vehicle then uses the internalized plan together with its sensor input to navigate. This combination of internalized plans and low level sensor based control makes the vehicle much more flexible and robust. In general, internalized plans take no extra computation, because all those states (grid points in Figure 9) must be checked to form the original plan.

In the multi-agent resupply domain, static dispatch orders are the equivalent of waypoint based plans in the single agent navigation example. In the multi-agent case, however, the drawbacks of static plans are even more severe. When a set of dispatch orders are created, certain assumptions are made about who needs what. Because there are many, possibly competing, agents in the environment, the situation changes. If an agent has no ability to sense the environment and make decisions, trips will be wasted. In addition, a static plan is only as good as the information from which it is constructed. A single agent has too limited a view to create a good static plan, and creating a coherent, central view from distributed inputs is also problematic.

We need to consider three requirements when formulating an internalized plan for a new problem domain,

1. Does the plan contain information that can be used to direct action?
2. Can an agent with a local view of the world correlate real-world sensory input to the plan in order to determine its applicability?
3. Can an agent with a local view of the world use sensory input to update the internalized plan?

Standard Plan with waypoints as subgoals



Gradient field as an internalized plan

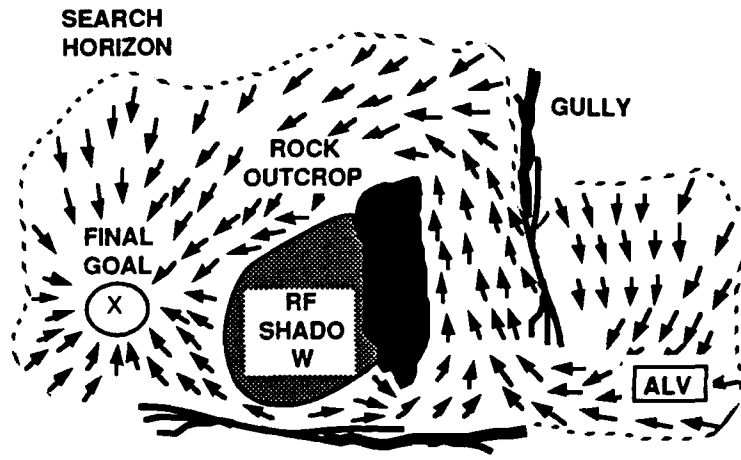


Figure 9: By representing a plan as a gradient field, instead of as a set of way points, an autonomous vehicle has more flexibility in using cues from the environment to direct its actions. Without this flexibility, time can be wasted achieving meaningless sub-goals.

For the resupply problem, we believe a graph flow representation is appropriate. A graph flow is simply the traversability graph with probabilities on each arc. Each agent can carry two graph representations, one for its own tasks and one for the entire team. For deterministic agents, an agent always takes the highest probability arc. Non-deterministic agents choose randomly biased by the probabilities. An agent can watch the traffic on the routes it uses and see if it matches with the team's plan. If the traffic does not match the plan then the agent can update its own graph.

3 Using the Mathematics of Economics

We have taken a number of ideas from economics in designing our cooperative autonomous agents test bed. We use these ideas because an economic system is an example of an environment where an overall goal is established by a central authority (a central bank), but the dynamics of the system are implemented by autonomous agents. In addition, much of the mathematics of economics deals with the flow of goods governed by supply and demand. Because supply and demand are quantities that an autonomous agent can sense, they are good candidates for representation in an internalized plan. Therefore the mathematics of economics are useful, even in domains where agents are not truly self seeking, economically minded entities.

Because we address *resupply* problems, where agents are carriers that repeatedly deliver goods to their "customers,"¹ it is natural to model the interactions of autonomous cooperating agents as a "market." This model may be decomposed into three related but separate issues:

1. What should the flows be for efficient delivery?
2. What is a good price and reward changing mechanism to govern the market?
3. What is a good (individualistic) response mechanism for agents to react to changing market conditions?

Issue (1) concerns the "global planning" (optimization) of the system, issues (2) and (3) concern the two dual (in the sense of mathematical equilibria programming) faces of "engineering" and "execution" of the plan. The first subsection describes a variant of simulated annealing to determine a good set of flows on the graph. The second subsection gives two different price setting rules that will guarantee smooth gradual changes in agent behavior as the situation varies away from that expected when setting the initial flows.

¹For instance, they may deliver materials to the factory workers, or they may deliver supplies and ammunition to military units.

3.1 Setting Prices

In order to formulate a plan that can serve as a resource for coordinating a team's activities, the individual agents are replaced by a "market," a single hypothetical super-agent aggregate of all agents. The market responds to a set of prices and rewards by returning an efficient flow. In this sense, the market is just another function; the prices and rewards are the input, the flows are the output. The response is "greedy" as it produces a flow that optimizes the market's rate of return. We give a nondeterministic Bayesian method, which is capable of returning the approximate market response for a very general class of typical nonlinear optimization problems arising in this setting. This approach seems to be justified by the great generality of these problems (general nonlinear complementarity problems) and, of course, when warranted by the special circumstances, the Bayesian method can be replaced by an appropriate direct method: linear complementarity techniques, convex programming, separable convex or quadratic network flow programming, or linear network flow programming.

Assume that there are N available agents, that can transport K distinct commodity types on a graph. Let $f = [f_{ij}^k]$ denote the vector (matrix) of flows (units/second), where f_{ij}^k denotes the flow of commodity $k = 1, 2, \dots, K$ from node i to node j . If we denote by f_{ij}^0 the flow of empty carriers across the arc (ij) , then we can define

$$\phi_{ij} = \sum_{k=0}^K f_{ij}^k, \quad (1)$$

the total flow from i to j .

Let $c(\phi) = [c_{ij}(\phi)] \geq 0$ denote the vector of time delays (seconds), where $c_{ij}(\phi)$ denotes the time required to traverse the arc (ij) from node i to node j . We assume that all $c_{ij}(\phi)$ are increasing (more precisely, nondecreasing) in each coordinate, however, we do not assume separability, convexity, or any other additional structure. The time delay function c is the fundamental property of the network.

Note that $F_{ij} = c_{ij}(\phi) \cdot \phi_{ij}$ is the total (expected) number of agents traversing the arc (ij) (units), and that we have a feasibility of flows condition,

$$\sum_{ij} F_{ij} \leq N, \quad (2)$$

which needs to be added to the usual conditions of conservation of flows.

Let us now turn to our Bayesian simulated annealing computational method. Assume that $E(f)$ is an "energy" function, which incorporates the utility that f would bring to the customer, as well as a penalty for violating the conservation of flows conditions.

Simulated annealing is characterized by a "temperature" parameter $T \geq 0$. If E_1 is the energy of the incumbent solution f^1 , and if E_2 is the energy of the newly proposed flow f^2 , then we accept f^2 as the new incumbent with probability

$$P(E_2 - E_1) = \min \left\{ 1, \exp\left(-\frac{E_2 - E_1}{T}\right) \right\}. \quad (3)$$

Note that f^2 is always accepted when $E_2 \leq E_1$.

In addition to the energy function we also need a means for generating f^2 that is biased towards directions that have been shown to be promising. Therefore we will divide our discussion into two parts, the energy function and generating new flows.

3.1.1 The energy function

The energy function can be any function that is lower for flows that better serve the customer's needs. To capture the idea of the customer's needs, we associate with each node, an exchange function that gives the relative value of each commodity in terms of all the other commodities.

In a multi-commodity market, a single set of flows can accommodate a number of assignments of goods to agents. Therefore, to find the value of a set of flows we optimize over all possible assignments of goods to agents, or equivalently, all possible assignments of goods to the flows. Of course we can only achieve this optimal use of a set of flows if agents have the ability to exchange commodities *en route*.

If the user's utility for each product at each location is approximately linear, then it is best to associate with each node ℓ an exchange matrix δ^ℓ , such that $\delta_{ii}^\ell = 0$ for all i and ℓ . These matrices are the aforementioned prices and rewards, the inputs to the market response mechanism; δ^ℓ specifies the costs or payoffs of exchanging the item k for item m at the node ℓ , for instance, $\delta_{k,0}^\ell$ represents the payoff for selling a unit of the good k , $\delta_{0,k}^\ell$ represents the cost of buying a unit of the good k . Impossible or undesirable transactions can be represented by large costs. For a given flow f , define the supply vectors s^ℓ by

$$s_k^\ell(f) = \sum_i f_{i,\ell}^k, \quad (4)$$

the demand vectors d^ℓ by

$$d_k^\ell(f) = \sum_j f_{\ell,j}^k. \quad (5)$$

The usual conservation of flows is satisfied if and only if $e^T s^\ell(f) = e^T d^\ell(f)$ (e is a vector of all ones) at every node. However, rather than enforcing the conservation of flows (a difficult task when the flows are selected by a random mechanism), we allow flows to violate the conservation condition. A penalty

proportional to the excess flow is then incurred. Mathematically, this is handled by creating a dummy commodity $K + 1$, which can be either bought or sold at each node for a high price.

Let the residual flows at the node ℓ be denoted by $r^\ell(f) = (e^\tau s^\ell(f) - e^\tau d^\ell(f))$ and let

$$\delta_{i(K+1)}^\ell = \delta_{(K+1)j}^\ell = M^\ell \quad (6)$$

for all $i, j = 0, 1, \dots, K$. Typically, M^ℓ are large numbers. At each node, solve the transportation problem

$$\begin{aligned} T^\ell(f) &= \min_x \sum_{i=0}^{K+1} \sum_{j=0}^{K+1} \delta_{ij}^\ell \cdot x_{ij} \\ \text{subject to} \quad &\sum_{j=0}^{K+1} x_{ij} = s_i^\ell, \text{ for all } i = 0, 1, \dots, K \\ &\sum_{i=0}^{K+1} x_{ij} = d_j^\ell, \text{ for all } j = 0, 1, \dots, K \\ &\sum_{j=0}^{K+1} x_{(K+1)j} = \max\{0, r^\ell\} \\ &\sum_{i=0}^{K+1} x_{i(K+1)} = \max\{0, -r^\ell\} \\ &x_{ij} \geq 0, \quad \text{for all } i, j = 0, 1, \dots, K + 1. \end{aligned}$$

The solutions of these problems provide the locally best transactions for the given set of flows. Therefore

$$E(f) = \sum_{\ell} T^\ell(f) \quad (7)$$

is the linear energy (utility) that can be assigned to the flows.

3.1.2 Choosing new flows

The essential ingredient of the Bayesian approach is that the flows are generated randomly from (piecewise constant) random distributions that are periodically updated. Figure 10 illustrates the point generation.

Assume that g (short for g_{ij}^k) is a current piecewise constant probability density Bayesian prior. Then sample f_{ij}^k from this distribution using uniformly distributed $(0, 1)$ random numbers. The number first determines the appropriate piecewise linear segment and then it gives the actual location within the segment. For instance, if the random number is L , then determine j such that $\sum_{i=1}^{j-1} a_i \leq$

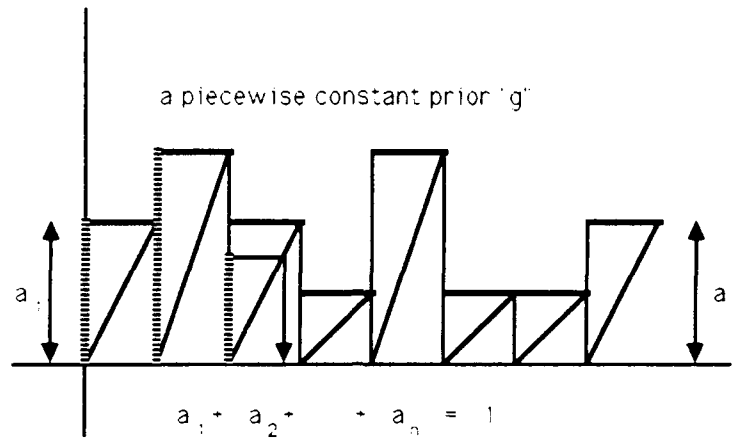


Figure 10: Example of a random point selection.

$L < \sum_{i=1}^j a_i$. This gives one the appropriate segment. Horizontally project $\bar{L} = L - \sum_{i=1}^{j-1} a_i$ to the (increasing) diagonal of the j -th rectangle, and then, vertically project the point down on the x axis. This gives you the sampled point.

The entire vector of flows, f , is so determined, each coordinate according to its own prior distribution. Then, f is rescaled to the unit length. These are the locations which are eventually charged with success or failure. Then, a random number, n between 0 and N is chosen according to its own prior (we can also maximize over the interval) and the flows are scaled (again) so that $\sum_{ij} F_{ij} = n$, (2).² This flow is submitted to the "oracle"; if the "oracle" accepts the proposal, then mark a success to the appropriate constant interval "bin", otherwise mark a failure. Finally, after enough statistics are collected, use the Bayesian formula to update the priors, g_{ij}^k . The "oracle" in our case is the energy function of Section 3.1.1, the optimal return for a set of flows.

3.2 Mechanisms of Price Modification

In the previous section, we described a method for generating a resupply plan based on the idea that a stable set of flows may be established, and that delivery vehicles should use this flow information as a guide in deciding how to reach an appropriate destination for the goods they carry. In setting up this kind of plan, we made use of a market model in which individual delivery vehicles could base their decisions on their expected cost for goods and travel versus their expected payoff upon delivery.

²Choosing $n < N$ corresponds to leaving some agents idle.

Once we find a set of flows that satisfies our expected delivery requirements, we still need to "enforce" this set of flows in the real market susceptible to random shocks and fluctuations. Well suited for the enforcement is the well known economic principle of *supply and demand*. In the resupply problem, this equates to the reduction in the payoff at nodes that are oversupplied, and an increase in payoff at nodes that are undersupplied.

Recall that each agent starts with an internalized plan, which is a set of flows it expects to see. An agent also gets input from the environment about how its deliveries are actually meeting customer needs. We characterize this in economic terms as the price received for the commodities. For agents to respond intelligently to a changing environment, prices must change as various nodes receive and consume supplies.

Intuitively, such price manipulation shall, in general, result in flow corrections that locally improve the flows. We could implement such a policy by any *ad hoc* rule that decreases the price as a node becomes over-supplied and increases the price as a node uses its supplies and requires more. However, it is not immediately clear whether, without some global guidance for implementing these price changes, the flows will vary in an orderly manner. The flow changes could just as easily be disproportionate, unstable, or otherwise problematic. Thus, in addition to the qualitative knowledge (that the prices need to decrease with respect to the supply delivery rates) we also need an analytical expression according to which we can modify the prices instead. Only then, through the market mechanism, can we truly control the flows of supplies.

Two analytical price-changing mechanisms are introduced here: the *quadratic-payoff economy*, and the *Cobb-Douglas economy*. Both price-changing mechanisms are intended for the general resupply problem (the general time-delay functions); however, for the purpose of this analysis, we assume that the time delays associated with arcs are constant. Additional simplifications are made for the sake of clarity (and tractability), but the basic principles should carry over to the the more general resupply markets.

3.2.1 Relationships and Contrasts between the Centralized and Distributed Planning

It is imperative to clearly distinguish between the various optimization and equilibria problems that we now encounter. Although, from the abstract point of view, the equilibria programming and mathematical optimizations are essentially equivalent, their relationships are often subtle. Let us analyze the resupply problem.

It simplifies the discussion to assume that there is only one type of commodity. Because all feasible flows (circulations) can be decomposed into simple circulations (conservation of flow), let us, for the analysis, assume that x_1, x_2, \dots, x_l are all simple circulations; associated with each such circulation is a (decreasing) payoff function $p_i(x_i)$, and a time delay constant c_i . Denote by $x, p(x)$, and c

the vectors of circulations, payoffs, and time delays, respectively. The *resupply market equilibria problem* arises from the following proposition.

Proposition 1. *If each delivery truck maximizes its own payoff rate, then these trucks effectively induce a vector of simple circulations, $x^* \geq 0$, such that for all $x_i^* > 0$*

$$\frac{p_i(x_i^*)}{c_i} \geq \frac{p_j(x_j^*)}{c_j}; \quad (8)$$

and

$$p_i(x_i^*) \geq 0, \quad (9)$$

and such that $c^T x \leq N$.

The equilibria problem is to find such an x^* . Note that (8) implies that if $x_i > 0$ and $x_j > 0$, then $\frac{p_i(x_i^*)}{c_i} = \frac{p_j(x_j^*)}{c_j}$. All trucks also must be used, $c^T x^* = N$, unless $p_i(x_i^*) = 0$ for all i . If

$$P_i(x) = \int^x p(s) ds, \quad (10)$$

the Stieltjes integral of $p_i(x_i)$, then we can associate the mathematical program

$$\max_{x \geq 0} \sum_{i=1}^{\ell} P_i(x_i), \text{ such that } c^T x \leq N \quad (11)$$

with the equilibria problem (8). It follows from the Lagrange conditions that the set of solutions of (8) contains the set of solutions of (10). Moreover, since $p(x)$ is a collection of decreasing functions, (10) is a strictly "convex"³ mathematical program, thus a unique x^* solves each problem.

There is another optimization problem associated with the market which is *not*, in general, equivalent to (8) and (10). However, the new problem is equivalent under certain assumptions and will be useful in subsequent derivations. Let $Q_i(x_i) = x_i \cdot p_i(x_i)$. Then the new problem is

$$\max_{x \geq 0} \sum_{i=1}^{\ell} Q_i(x_i) = x^T p(x), \text{ such that } c^T x \leq N. \quad (12)$$

³Actually, it is the equivalent minimization problem,

$$\min_{x \geq 0} - \sum_{i=1}^{\ell} P_i(x_i), \text{ such that } c^T x \leq N$$

that is convex.

The solutions of this possibly nonconvex problem gives the maximum total payoff rate achievable by the grand coalition of all trucks cooperating together.⁴ Denote a solution of (12) by \bar{x} .

Clearly, \bar{x} typically is a suboptimal solution of (10) and x^* typically is a suboptimal solution of (12). If x^* is the desired set of circulations, then we need to prevent a collusion of the agents—we either need to prevent it by some sort of an “antitrust law” (where the agents are explicitly prohibited to form coalitions), or we need to prevent it implicitly by creating a market such that (10) and (12) are solved by $\bar{x} = x^*$. Although it is not immediately clear whether this can be done, the latter alternative can be accomplished, since we have the control of the market mechanisms. We can, in fact, create markets which have some additional attractive properties as well. For instance, we can create such price-changing mechanisms that if the number of available trucks changes from N to αN ($\alpha \geq 0$), then the solution of the problem (10) (and (12)) changes proportionally from x^* to αx^* .

3.2.2 Quadratic-Payoff Economy

A simple change of variables, $x_i \mapsto c_i x_i$, allows us to assume that instead of the general time delay vector c , all time delays are the same, $c = e$. Price functionals of the quadratic-payoff economies are of the form

$$p_i(x_i) = b_i - a_i x_i, \quad (13)$$

where a_i and b_i are positive numbers. Therefore the components of the objective function of the program (11) are

$$P_i(x_i) = b_i x_i - a_i \frac{x_i^2}{2}, \quad (14)$$

and the components of the objective function of the program (12) are

$$Q_i(x_i) = b_i x_i - a_i x_i^2. \quad (15)$$

Note the close relationship between the functions,

$$Q_i(x_i) = P_i(x_i) - a_i \frac{x_i^2}{2}. \quad (16)$$

Now, assume that g is the desirable set of (goal) circulations and note that, by definition, $e^T g = N$. Without loss of generality, assume that g is positive, because it suffices to work with the support of g otherwise.

⁴Some payoff structures can be exploited by the teams of agents if some agents are willing to sacrifice their own earnings in order to improve the overall earning of the group. The earning of such agents decreases but this is more than adequately compensated by the increase in the earnings of the other agents.

Lemma 2. *If*

$$p_i(\alpha g_i) = p_j(\alpha g_j), \quad \text{for all } i, j, \text{ and all } \alpha > 0, \quad (17)$$

and if

$$a_i g_i = a_j g_j, \quad \text{for all } i, j, \quad (18)$$

then αg is the unique solution of

$$\max_{x \geq 0} \sum_{i=1}^{\ell} P_i(x_i), \text{ such that } e^T x \leq \alpha N \quad (19)$$

and

$$\max_{x \geq 0} \sum_{i=1}^{\ell} Q_i(x_i), \text{ such that } e^T x \leq \alpha N \quad (20)$$

for any positive α .

Proof: Since both mathematical programs, (19) and (20), are strictly "convex" (and bounded), they have unique solutions. If $p(x)$ satisfies (17), then αg is the unique solution to (19). Then, in view of (16), note that for all i ,

$$\frac{dQ_i(x_i)}{dx_i} = p_i(x_i) - a_i x_i. \quad (21)$$

From (17) and (18) thus follows that if $x^* = \alpha g$, then

$$\frac{dQ_i(x_i^*)}{dx_i} = \frac{dQ_j(x_j^*)}{dx_j}, \quad \text{for all } i, j. \quad (22)$$

Verify the Lagrange conditions of (20) to complete the proof. \square

All that now remains is to notice that if for all i we set $b_i = B$, positive constant, and if $a_i = \frac{1}{g_i}$, then $p(x)$ satisfies both functional conditions, (17) and (18). Thus we have succeeded in creating a market of the desirable type.

3.2.3 Cobb-Douglas Economy

Let all basic assumptions about the simple circulation market be as in Section 3.2.2. Let a be a positive vector. In Cobb-Douglas economy

$$\max_{x \geq 0} \prod_{i=1}^{\ell} x_i^{a_i}, \text{ such that } e^T x \leq \alpha N. \quad (23)$$

This is clearly equivalent (after taking a log of the objective function) to

$$\max_{x \geq 0} \sum_{i=1}^{\ell} a_i \log(x_i), \text{ such that } e^T x \leq \alpha N. \quad (24)$$

Note that for each i

$$P_i(x_i) = a_i \log(x_i), \quad (25)$$

$$p_i(x_i) = \frac{a_i}{x_i}, \quad (26)$$

and

$$Q_i(x_i) = a_i. \quad (27)$$

It follows from the Lagrange conditions that (24) is uniquely solved by $x^* = \alpha c$. Since all Q_i are constant, any feasible x trivially solves the "grand coalition problem" (12). Thus it suffices to set $a = g$ to achieve the desirable results. \square

3.3 Decomposing Graphs into Cycles

We can utilize the notion of a cycle for two different purposes. First, autonomous agents can use cycle decompositions of desired flows as bases for their reasoning. On the basis of payoff on each cycle, individual agents choose (and switch between) cyclical routes along which they deliver supplies. We can also use cycles to generate feasible flows, a technique which under certain simplifying assumptions greatly reduces the number of dimensions of the space that needs to be sampled.

A vector of feasible flows in the resupply problem is a *circulation*. That is, if $\phi = (\phi_{ij})$ is a vector of (aggregated) feasible flows, then it can be decomposed into a sum of *cycles*, (see Figure 11). The feasible flows then result from the superposition of the cycles. A cycle is *simple* if no vertex is repeated until the cycle completes and the flow is the same on every arc in the cycle. All cycles in Figure 11 are simple and, in fact, any circulation can be decomposed into simple cycles. Note that in breaking a flow into cycles, each arc participates in many different cycles.

The decomposition of a feasible flow into simple cycles can be achieved by this method. Follow the given cycle until a first repetition of a vertex occurs. That completes a simple cycle. Subtract this cycle from the flow and repeat the procedure until the decomposition is completed. Figure 12 gives an idea how this is done. In more generality, given a circulation, remove from the graph all arcs with zero flow. Then walk randomly (on the arcs with positive flow) until the first time a node is repeated. This determines some simple cycle. Let the least flow on the arcs of this cycle become the cycle's flow. Subtract this flow from the circulation and, again, remove from the graph all arcs with zero

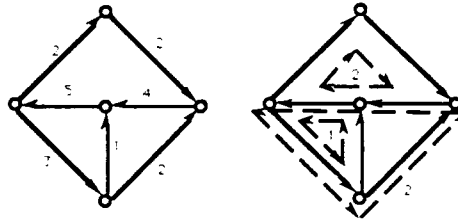


Figure 11: Example of a decomposition of a feasible flow into cycles. Numbers on the arcs indicate the amount of flow on the arc. In the second graph, dashed lines indicate circular flows of constant amount.

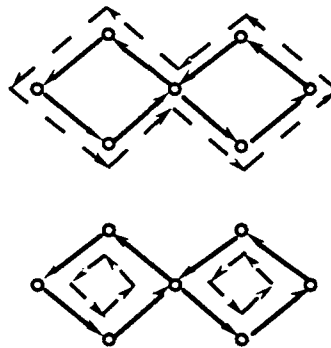


Figure 12: A cycle decomposed into simple cycles.

flow. The remaining flow is also a circulation. If the resulting graph contains no arcs, then we are done, otherwise repeat the procedure. Since at every stage at least one arc becomes zero, if A denotes the set of arcs, then this algorithm terminates in at most $O(|A|)$ stages.⁵

4 Knowledge Acquisition: Evaluating the Appropriateness of Our Model

In this section, we present an analysis of the insights gained from discussions with Captain Dennis Szydloski and Captain Gary Krzysnik at Fort Knox. During two days of meetings, we discussed a range of topics relating to the use of cooperating autonomous agents within the military. Our primary emphasis was on the development of suitable scenarios for resupply, but we also discussed the possibilities of developing autonomous tactical vehicles which might be able to cooperate with their human counterparts on the battlefield.

⁵The outlined algorithm can be significantly streamlined.

From our discussions about military resupply, we found that the overall problem presents many interesting and challenging complexities. The army categorizes supplies into ten different classes. Of these classes, most items are dispatched in response to specific orders or requests. However, classes I, III, and IV, which include rations, fuel, and ammunition, all tend to be allocated on the basis of a combination of future consumption estimates and updates of current demand. The most interesting case is ammunition resupply, in which a required supply rate (RSR) is established at regular intervals to identify the amounts of ammunition needed to sustain operations. This figure, however, must be balanced against a controlled supply rate (CSR). The controlled supply rate is established on the basis of availability, facilities, and transportation for a given period. Often, it is possible for a battalion's CSR to be less than its RSR. When this is the case, the overall allocation of supplies is established by the division commander and is then re-allocated by each subordinate commander.

We believe that because rations, fuel, and ammunition all have relatively stable rates of consumption, the resupply problem for these commodities may be suitably represented as a flow problem which can be solved by autonomous agents. Our representation of this problem, however, must take into account the dynamic nature of the battlefield environment. Resupply efforts must function smoothly in the presence of troop degradation and rapid troop movement. If autonomous agents are to assume a resupply role, they will need the ability to compensate for the loss of their team members and the ability to function effectively even if their commanding headquarters is incapacitated. From this perspective, our flow representation discussed in previous reports is a good candidate solution.

Despite its appeal from the standpoint of adaptability, we found that our flow representation does not fit easily within the military infrastructure. In our flow model, it was assumed that agents could dynamically choose their paths to delivery and supply points from a variety of alternate paths in a road network. Instead, we were told that supply routes are typically fixed and heavily defended so that there is rarely the desire to choose alternate paths. In addition, we learned that supply quantities are often carefully pre-packaged in the exact quantities needed for their specific destinations, allowing delivery vehicles to unload and get out of threatening areas as quickly as possible. In our autonomous agent approach, we provide opportunism by allowing agents flexibility with regards to when and where they may deliver their cargo. By doing this, supplies cannot easily be pre-packaged for delivery to specific sites.

Another aspect of resupply which complicates matters from the standpoint of autonomous delivery is that battalions typically use two different methods to obtain supplies. These are: supply point distribution, and unit distribution. In supply point distribution, a unit uses its own vehicles to go to a central supply point and pick up supplies. In unit distribution, supplies are delivered to units by transportation assets other than their own. Under our scenarios for autonomous supply delivery, we assumed a model which was purely based on

unit distribution. To allow for both types of distribution, autonomous agents may have to function cooperatively with the human agents that are performing their own supply point distribution tasks.

It may also be worth investigating whether the use of autonomous agents for resupply might alter some of the above restrictions and methodologies, thereby allowing our more dynamic supply model to be implemented. For example, if autonomous agents are more expendable than their human counterparts, it may be possible to allow them to disperse and take advantage of a variety of supply routes. While this would be harder to defend, it would be more reliable since cutting off any single supply route could not stop the flow of supplies. Similarly, the urgency to get supply vehicles out of high risk territory may be reduced if these vehicles are unmanned. If a delivery vehicle can spend the time to deliver supplies to each fighting vehicle, then the overall risk to humans can be greatly reduced.

In order to properly evaluate the possibilities for doctrinal change afforded by autonomous delivery agents, it would be necessary to have a highly realistic simulation of the resupply problem so that existing techniques could be compared to the alternatives. Often, existing doctrine is established on the basis of many complex constraints that are not always apparent to the uninitiated. Simulation can often reveal the effects of these constraints, but only if they are correctly included within the simulation model. For the military resupply problem, one of the most important and yet most difficult to model factors is the interaction between man and machine.

5 Using SIMNET as a Testbed

While at Fort Knox, we also investigated the possibility of using the SIMNET simulation environment as a testbed for our multi-agent research. SIMNET provides a unique environment where military teams up to the size of a battalion can train for battle without using real military vehicles. The simulation creates sufficiently realistic sensory input for human trainees that they can learn a great deal about how to function as an effective team as they engage in simulated battle scenarios. We explored the possibilities of either integrating a team of resupply agents or individual fighting agents into the SIMNET environment.

SIMNET is structured as a highly distributed network of simulators. Each simulator is manned by a human crew, and the actions of the crew are transmitted through the network so that the world representation is consistent between all simulators. The visual displays presented to each tank crew are simple, but sufficient to provide a good sense of terrain and other vehicles. Because of its highly distributed nature, it is possible to insert a completely automated player into the SIMNET network. This is currently being done by BBN for their simulation of semi-automated opposing forces.

In exploring the possibility of automating the resupply task within SIMNET,

we found that currently, supply trucks are simply made to appear at destinations designated by a supply commander. To be able to study interesting supply problems, the simulation would need to support the operation of supply vehicles through all phases of their delivery task. Another problem with simulating supply is the overall duration of a SIMNET training session. Rarely does a training session last long enough for resupply to become an important issue in the training exercise.

SIMNET would be especially useful in the study of the incorporation of autonomous agents into a platoon. SIMNET provides the opportunity to study the substitution of an autonomous system for a tank crew. The issues of interest would revolve around what capabilities would be needed from such a system to allow the human platoon members to fight effectively. The autonomous system would clearly need to be able to interpret and cooperate with the platoon leader. It would also need to perform many sensing and decision-making tasks on its own. By using SIMNET, it may be possible for military planners to explore different roles of autonomous systems before committing to large expenditures on hardware.

6 Conclusions

We feel that the approach developed during the first year is technically sound. It provides a framework for creating and executing flexible resupply plans in a changing environment. However, after our knowledge acquisition at Fort Knox, we found that our initial formulation was simple compared with the complexities of real resupply problems. To demonstrate the feasibility of this approach would require substantial knowledge acquisition to create the mathematical formalisms, and it would require extensive coding and testing of algorithms. We view this as a high risk endeavor.

A better application of this technology would be the Semi-Automated Forces (SAFOR) in SIMNET. Such a shift in emphasis serves two purposes. First we can develop multi-agent planning algorithms that can be used in the long term for autonomous and semi-autonomous combat vehicles. Second, in the short term, we can use our algorithms to decrease the number of humans needed for opposing forces in a SIMNET exercise.

7 References

Payton, D. W. (1988). Internalized Plans: A Representation for Action Resources. *Workshop on Representation and Learning in an Autonomous Agent*, Lagos, Portugal:

Daily, M., Harris, J., Keirse, D., Olin, K., Payton, D., Reiser, K., Rosenblatt, J., Tseng, D., and Wong, V. (1987). *Autonomous Cross-Country Nav-*

igation with the ALV. *DARPA Knowledge Based Planning Workshop*, Austin,
Texas: