Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Cooperative Control for Multiple Autonomous UAV's Searching for Targets[1]

Matthew Flint [2] Marios Polycarpou [3] Emmanuel Fernández-Gaucherand [4]
Dept. of Electrical and Computer Eng. and Computer Science
University of Cincinnati,
Cincinnati, Ohio 45221-0030, USA

Abstract

The work presented here is in the area of decision and control for autonomous unmanned aerial vehicles (UAV's). Specifically, we formulate the problem of generating near-optimal trajectories to follow in order for several UAV's to cooperatively search for targets in a given area for which some a priori data about target distribution is available. An algorithm that utilizes a model of the cooperation is developed and a dynamic programming implementation is presented as a solution to this problem. Results from simulations are provided to illustrate the usefulness of this approach.

1 Introduction

Directing autonomous agents to behave in an "intelligent" manner provides for a very interesting problem, especially in cases where there exist many constraints on the control of the agents. In the particular case of multiple autonomous Unmanned Aerial Vehicles (UAV's) searching for targets in an uncertain environment, this is certainly true. While the problem is a "search" problem, there exist many factors that set this problem apart from many of the classical search problems, such as those discussed in [4]. For example, optimal paths are desired, rather than an optimal allocation of effort, and those paths are constrained by, e.g., the limited ability of UAV's to make high-G turns. Additionally, the presence of multiple vehicles brings the concept of cooperation among the vehicles to the forefront, since it is to be expected that a team of vehicles acting in concert would perform better than a mob of vehicles acting independently.

The work presented here is part of a growing body of research that is aimed at developing cooperative control algorithms that will allow a team of UAV's to fly missions without direct human intervention, and in the presence of uncertainty [5]. Polycarpou, et al., in [7], develop a general framework for this problem.

This paper formulates the problem of multiple UAV's that must autonomously generate paths over the terrain such that, given some a priori information, the maximum number of targets in the environment can be positively identified. A discrete time stochastic decision model is formulated for this problem, which is then implemented with a Dynamic Programming algorithm. This approach has been successfully applied to other UAV problems, as in [6]. To overcome the computational complexity issues inherent in such an implementation, approximations are used to produce efficient sub-optimal paths for the vehicles to follow.

There are many applications for UAV's like those motivating this work: emergency vehicles searching for lost or stranded persons in dangerous environments; autonomous munitions, searching for military targets that will be subsequently attacked; or unmanned intelligence gathering aircraft. In all cases, it is desired that the vehicles respond to their environment intelligently, where they begin with some knowledge of their environment, which they utilize, and then cooperate with the other vehicles in order to perform a better search.

This paper expands on the methods and ideas presented in [3]. While [3] concentrated on general ideas, this paper presents a detailed treatment of the foundations and implementation of the model and solution. Also, the usefulness of the formulation is increased by including, explicitly, targets in the cost function and as part of the measure of effectiveness.

2 Problem Formulation

2.1 The Vehicles

The goal of each UAV in the search problem is to move over the environment such that, at the end of the search, the maximum number of new targets has been identified. A vehicle must therefore plan a *local* path over the environment to meet this *global* goal.

[1]This work was partially supported by DAGSI (Dayton Area Graduate Studies Institue) and DARPA under the MICA (Mixed Initiative Control of Automa-teams) project.

[2]flintmd@email.uc.edu

[3]polycarpou@uc.edu

[4]emmanuel@ececs.uc.edu

The path planning problem is discretized in time by allowing the vehicle to only make decisions at discrete time intervals. These intervals will hereafter be referred to as "time steps". The ability of the vehicle to make turns is constrained by a discretization in space by only allowing the vehicle to make a limited number (m) of choices at each time step (e.g. for $m = 3$ these choices may be: turn 15° left, go straight, or turn 15° right.)

It is assumed that the vehicles can communicate over a wireless channel. Each vehicle sends location and heading information about itself to the other vehicles. The communication bandwidth is not unlimited, however. This is modeled by allowing the vehicles to communicate only every b time steps, where a higher value of b represents a more limited channel. So, the vehicles will be making b decisions per communication step, and thus $b - 1$ steps with progressively outdated information. It is also assumed that each vehicle is equipped with a sensor. This sensor projects a window out in front of it in which the vehicle can detect the environment below. The sensor has some probability of detecting a target given that the vehicle encounters that target, i.e. that a target lies within its sensor window.

For simplicity, a typical vehicle is assumed to fly at a constant velocity and altitude, and to be able to avoid colliding with other vehicles. The lifetime of the vehicle, determined by the amount of fuel it carries, will be N time steps. Additionally, at each time step the vehicle must also choose a path that is at least q steps ahead.

2.2 The Environment

The environment is given by a bounded search area, about which some a priori information may be available. A vehicle carries a representation of the environment in its memory in the form of a cognitive map. The information on this map is what the vehicle bases its decisions on. As a vehicle's sensor passes over the terrain, a corresponding area of the map is updated to reflect the information gained from the vehicle's sensor.

Stationed in the environment are targets, the number of which is not known, and the location of which is hinted at by the a priori data. The targets are, at least initially, assumed to be stationary and that there is no correlation between the locations of targets (i.e. knowing the location of one target provides no information about the location of any other target.)

The a priori data about the environment is assumed to be given as information about the likelihood of targets being located in different regions of the environment. For any region r_1, let $C_{r1} \in [0, 1]$ be the normalized measure of the value of searching the region such that, for any regions r_1 and r_2, if $\frac{C_{r1}}{C_{r2}} = X$, then there is X times as much probability of there being a target in region r_1 than region r_2, assuming they are of the same area. For example, it could be known that, given an equal area, it is 10 times more likely to find a target on the road than in the forest.

Let A_r be the area of any region r (of which there are n total), and $A_x (= (A_1 + A_2 + \ldots + A_n))$ be the area of the whole map. Then, let

$$Q_x = \sum_{r=1}^{n} (A_r \times C_r) \qquad (1)$$

so that now the probability perceived by the vehicle of a target being in region r can be calculated by

$$P_r = \frac{A_r \times C_r}{Q_x} \qquad (2)$$

A vehicle's map is split into a grid of equal sized unit areas, which are termed *points*, where the area of each point (A_p) is the resolution that the vehicle will use on its map, and thus within the area of each point p, there is assumed to be a constant C_p. The cognitive map aboard each vehicle then stores the C_p for every point.

3 Main Results

3.1 The Decision Model and DP Solution

In order for the vehicles to plan their trajectories, they need to know information about the state of the environment. The true state of the environment, after discretization, is represented by: (1) a map like those that the vehicles carry, except that it is updated by every vehicle at every time step (this will be referred to as the true map); and (2) by the locations and headings of the vehicles. This true state is denoted by x_k^*. The state, as perceived by vehicle i, is denoted as x_k^i, or if the vehicle index is assumed, x_k. Under ideal conditions, every vehicle will perceive the state as being the true state ($x_k^* = x_k^i \ \forall i$.)

The choice of which path to travel for vehicle i at time k is the decision, or control u_k^i (u_k if the vehicle index is assumed), and $u_k^i \in U$, where U is a set of size m and contains the choices that the vehicle can take (for example: turn 15° left, go straight, or turn 15° right.)

The points that vehicle i's sensor would pass over (and would be updated on its map) as the vehicle travels from state x_k^i to the next state by taking the decision u_k^i are denoted by z_{x_k,u_k}^i (or z_{x_k,u_k} if the vehicle number is assumed.) Note that these points depend only on what path the decision making vehicle chooses to take (u_k^i), and not on any other vehicle.

Since the goal of the vehicles is to find targets, and the targets are distributed according to some perceived distribution, the gain function should maximize the perceived probability of finding a target. Gain is thus defined as the reduction in probability of there being an undiscovered target in a searched area. Our objective would be to maximize the expected gains over the horizon N for all the vehicles [2].

For v vehicles, the total gain at a time step k is a sum of the gains of the individual vehicles:

$$g(x_k, u_k) = \sum_{i=1}^{v} g_i(x_k, u_k) \tag{3}$$

Under the assumption of unlimited computational power, the one-step gain function for an individual vehicle is defined as

$$g_i(x_k, u_k) = \sigma(z^i_{x_k,u_k}) \tag{4}$$

where $\sigma(z^i_{x_k,u_k})$ is a function that returns the gain from searching the points in $z^i_{x_k,u_k}$, assuming that no other vehicle interferes. Interference is defined as having two or more vehicles searching the same points, when it would be better if the vehicles were searching different points.

Let P_p be the probability of there being an undiscovered target in point p before searching the point. Let P'_p be the probability after the search (i.e. the probability of there being a target at that point and that it was not found). Let ρ be the probability of finding a target on one search of the point (given that the target was at that point). Using conditional probability,

$$P'_p = P_p(1-\rho) \tag{5}$$

Thus, the gain from searching a point (p) is

$$\sigma(p) = P_p - P'_p = P_p - P_p(1-\rho) = P_p - P_p + \rho P_p = \rho P_p \tag{6}$$

and the gain from searching any arbitrary collection of points z_a is

$$\sigma(z_a) = \rho \sum_{p \in z_a} P_p = \frac{\rho A_p}{Q_x} \sum_{p \in z_a} C_p \tag{7}$$

Let J_k be defined as what is commonly called a "cost-to-go" function [2], or, since the formulation is in terms of gain instead of cost, a "gain-for-going" function, from time step k to N. The problem can be solved, under the assumption of unlimited computational power, by letting the final gain (at step N) be $J_N = 0$ (no bonus for being in any particular state at time N), and solving the DP recursion from time steps 1 to N:

$$J_k(x_k) = \max_{u_k \in U}(\{g(x_k, u_k) + J_{k+1}(f(x_k, u_k)\}) \tag{8}$$

where $g(x_k, u_k)$ is the one-step gain function.

Therefore, in the idealistic case where the vehicles have unlimited computational power, it would be possible to treat the problem in an open-loop framework, which could be solved before the vehicles are put into flight by letting the vehicles expand not only their own search paths, but also those of every other vehicle. Each vehicle could thus find the globally best path not only for itself, but also for each other vehicle. Thus, at any one time step, they would know where the other vehicles would be and would be able to plan a path such that they would not interfere.

3.2 Refining the Model and Solution

The implicit cooperation detailed above will cease to function when the unlimited computation assumption is removed, as it would for a real-world implementation. This results from the fact that every vehicle can no longer expand every possible path of every vehicle, and is thus going to have to rely on other methods to avoid interfering. To implement the proposed scheme in a realistic setting, then, there has to be some simplifying assumptions.

Since the state of a vehicle is a path on a map, it is natural to view this as a line on the search map extending behind the vehicle, connecting all the points the vehicle has traveled, and to view the planning of the vehicle as a tree starting from the vehicle's current location, and to view each branch of the tree as a potential state.

To address the problem of expanding the search tree out to the end of the vehicle's own lifetime, N, a limited look-ahead policy is used, in which rather than computing the state to the end of the vehicle's lifetime (a tree of depth N), it replaces the cost-to-go at state $k+r$ (J_{k+r}) with the final cost (J_N) [2]. This allows the system to calculate the paths generated by only going to a search depth of r, and to choose the best path visible in this sub-problem. The size of the abbreviated problem is in general much smaller than the size of the original problem. Thus r can be chosen such that the problem is tractable.

In order to give the vehicles a method by which to predict the other vehicle's behaviors without having to explicitly calculate the other vehicle's moves, the other vehicles are modeled as *stochastic elements*, where a random quantity w_k is used to represent the loss in gain at time $k+q$ (compared to what was expected when the decision is made at time k) because of interference by another vehicle. The vehicles can then use the expected result of this w_k to plan where to go to avoid undue interference. The distribution of w_k depends on z_{x_k,u_k}, since different areas have different likelihoods of vehicles scanning them.

The gain at each step is then what one would get if no interference were present minus the amount of gain one would lose from another vehicle interfering,

$$g(x_k, u_k, w_k) = \sigma(z_{x_k,u_k}) - w_k(z_{x_k,u_k}) \qquad (9)$$

This modified gain is then used in the DP recursion (Equation (8)) to determine the expected optimal path.

The amount of interference a vehicle expects at a point p of the map is a function of ρ (since an interfering vehicle will have reduced the value of p by ρ), $\sigma(p)$ and the probability of another vehicle interfering (i.e. scanning point p before this vehicle does), which is denoted as $\Pr(p)$. So

$$E(w_k(p)) = \rho\ \sigma(p)\ \Pr(p) \qquad (10)$$

Summing all the points in an area z_{x_k,u_k} gives

$$E(w_k(z_{x_k,u_k})) = \rho \sum_{p \in z_{x_k,u_k}} \sigma(p)\ \Pr(p) \qquad (11)$$

It is important to note that the interference factors are not symmetric between interfering vehicles, since the first vehicle to search a region will receive full gain, even if the second vehicle will not.

Since the vehicles' behavior is governed by a known algorithm, the distribution of probability of where any particular vehicle will be is known to have some structure in space, and the exact shape of this structure is formed from the constraints on the maneuverability of the vehicle. Calculating $\Pr(i)$ for all $i \in z_{x_k,u_k}$ exactly would require each vehicle to expand every other vehicle's planning tree, and assign a likelihood to each point on the map that would be scanned by traveling that path based on the probability of the vehicle traveling that path. Where that vehicle will go is based on information that the planning vehicle may not have access to, since the vehicle's maps may not be the same, so the planning vehicle must take an estimate of where the other vehicle will be.

So, an approximate structure is formed by taking several several spatial regions where each region represents a certain constant probability of the vehicle causing interference. Figure 1 shows a typical situation when this process would be used. The vehicle with the eye represents where the planning vehicle will be in q steps, and thus, where its planning will start from. The tree of possible paths extends from it, and the best path must be chosen from this tree. The vehicle plans its moves in sequential order, so from left to right the figure shows the decision process in time. The vehicle will travel straight in the absence of any other factors, and the heavy line shows the path the vehicle has chosen. Each of the shaded regions extending from the other vehicle represents the regions where interference is estimated

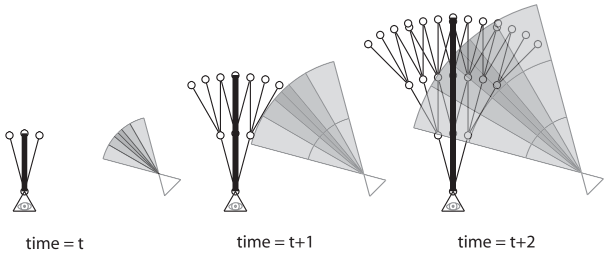

Figure 1: A vehicle is planning where to go based on another vehicle's probable location.

to be able to happen. Note that the approximation region for this grows as the planning vehicle extends its planning in time to take into account the fact that the vehicles are moving.

Thus cooperation is achieved without the need for a vehicle to expand any planning trees but its own. The cooperation is of a passive sort, since the vehicles do not negotiate. Instead, they use the communication channel only to send knowledge of their locations and headings. The structure of the regions, also, is flexible, and the values can be tuned for better results or to incorporate new information, with the additional possibility of online adjustment.

Following [2], a label correcting method was chosen to do the search along the tree of possible choices a vehicle could make. The algorithmic solution used falls within the so-called "best first" method.

4 Simulation Results

A set of simulation studies was done to check the effectiveness of the algorithm in finding targets in a scenario that vehicles might face (e.g. as part of a military strike). This is compared to the same environment searched by a standard search, which means a search pattern that is decided upon a priori, e.g., a Zamboni search; and also against the same algorithm, but with the communication channel placed under more severe restrictions. What this will show is that the proposed algorithm can perform better than a standard search, and also that improved cooperation also improves the search.

In this example, the vehicles have 3 decisions at each time step ($m = 3$):{ turn 15 degrees left, go straight, or turn 15 degrees right}. The search depth is 6 times steps, so that is how many steps ahead the vehicles are planning. The vehicles move at a constant speed of 5 units distance per time step, so they are planning 30 units distance ahead of where they know they will be with certainty. The vehicles are also planning with a 3 step buffer ($q = 3$), so they are planning a total of 45 units ahead of where they are actually located. There are 4 vehicles in this scenario, and the chance of detecting a target given an encounter is 50% ($\rho = 0.5$). The time between communications is 2 time steps ($b = 2$).

A priori data can come from both the terrain and from other sources of information. The default type of terrain in this example scenario is the white space on the map (as seen in the "background" of Figure 2.) This represents flat areas about which nothing remarkable is known. Information from other types of terrain comes in the form of two features: a lake and a road. It is known that the targets cannot float, thus there is 0% chance of finding a target on the lake (the circular region in the middle of the figure.) However, it is known that targets will be more likely to be found on the road (the narrow band running vertically along the figure.) An equal sized comparison of the white space area is 90% as likely to contain a target as the road. Other information has also been provided. In this example, the a priori data is from satellites and previous searches. The hatched area in the upper left indicates an area where satellite surveillance has determined that it is only 10% as likely to find a target in this region as in the unshaded region. The hatched region in the lower right shows an area that has been subjected to a previous search, and thus any terrain in this area is 50% ($= \rho$) as likely to contain any new targets as the same type of terrain in the unhatched area. The cross-hatched region in the center of the map shows an area where a satellite has found increased target activity. Thus, it is known that there is twice as much chance of finding a target in this area as in the corresponding areas in the unshaded region (if comparing equal sized regions.)

The search depth is 6 time steps, so the approximation region is also six levels. For each test run, the vehicles have the same starting position. The targets are randomly assigned to the environment according to the a priori information. Then the vehicles search the area 10 times, recording the number of targets encountered, and then the number of targets positively identified. It is possible for a vehicle to encounter a target (i.e. to have it appear within its sensor window) but not to identify it. The chance of this happening is governed by ρ. This process is repeated 1000 times for differing maximum lifetimes.

Figure 2 shows the result of running the algorithm shown in this paper in the given environment. Figure 3 displays the standard search that it is compared against. In both cases, the vehicles begin in the bottom left hand corner, with each vehicle's path represented by a line. The standard search is in this case a zamboni search, so called because it resembles the path a zamboni machine takes over an ice rink [1].

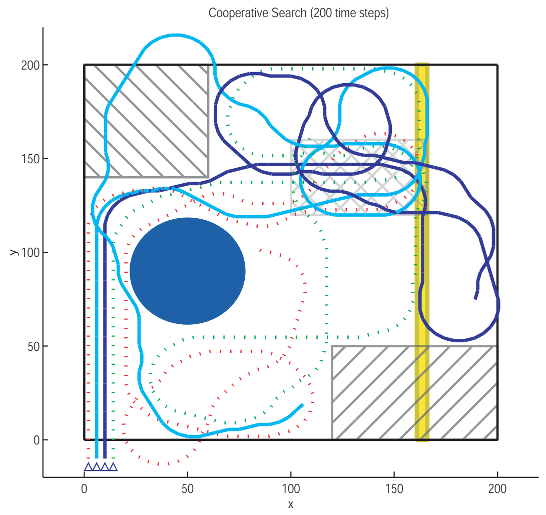

Figure 2: Vehicle's movements for the proposed algorithm's search.

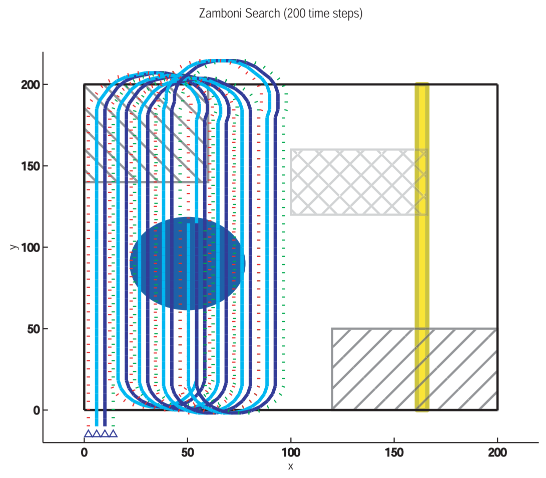

Figure 3: Vehicle's movements for the ZAMBONI search.

Note that in this example one can gain a general view of the allocation of effort for each method. The zamboni search covers the lake and the region where the satellite information has determined that there is little chance of finding a target. On the other hand, the cooperative algorithm generally allocates its effort to areas where there is better chance of finding targets.

The number of targets found during the simulations is shown in Figures 4 and 5. Figure 4 shows how the algorithm proposed in this paper fares against the zamboni search, from Figure 3. The solid lines show the number of targets positively identified by each algorithm. The lines marked with squares are the results from the proposed algorithm, while the lines marked with the circles are the results of the zamboni algorithm. The dashed lines show the number of target encounters of each algorithm, where an encounter is where a vehi-

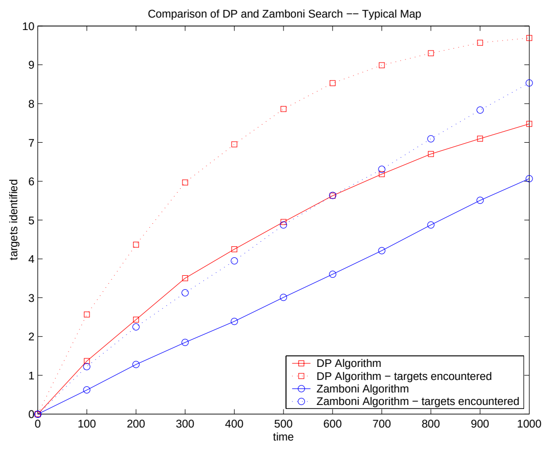

Figure 4: Comparison of Effectiveness of Algorithms.

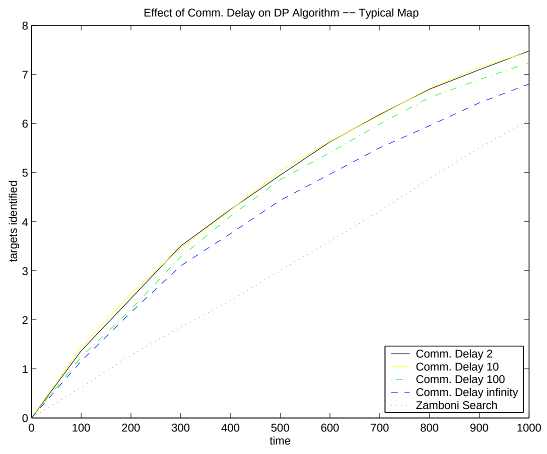

Figure 5: The Effects of Cooperation (or the lack thereof).

cle's sensor searches a point with a target, whether or not the sensor actually detects the target. This figure shows how much better the proposed algorithm can perform than a standard search pattern. As can be seen, the proposed algorithm is much more efficient at detecting targets (the solid lines), and at covering the areas with targets (the dashed lines.)

Figure 5 shows the effect of communication (or the lack of it.) Since the cooperation of the vehicles in this scheme is passive, the amount of communication directly effects the amount of cooperation that the vehicles exhibit, and one would expect that decreased communication would lead to a worse search performance. The same algorithm was run several times in the same environment, except that the communication batch delay was increased until finally, at a batch delay of infinity, the vehicles were not permitted to communicate at all, and thus, are not allowed any form of cooperation. As can be seen from this diagram, deceased communication also decreased the average number of targets found on the search. However, note that even with tight restrictions on communication, the algorithm can still perform very well compared to the zamboni search. Additionally, the results here justify the assertion that cooperation is beneficial to the overall search, and give a rationale for trying to improve the cooperation, in order to perform a better overall search.

5 Conclusion

The performance of autonomous UAV's searching for targets has been shown to benefit from a decision and control scheme that creates a model in which gain is based on maximizing the expected number of targets found given some a priori information, and in which a feasible method of cooperation is achieved through considering other vehicles as stochastic elements. This scheme lends itself to an implementation using a dynamic programming solution. Even though the results of removing the assumption of unlimited computation produce a necessarily sub-optimal result, the paths generated by the algorithm are still quite efficient at finding targets, as was shown in the simulation study.

References

[1] V. Ablavsky and M. Snorrason. Optimal search for a moving target: A geometric approach. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000.

[2] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Belmont, Massachusetts, 2nd edition, 2000.

[3] M. Flint, M. Polycarpou, and E. Fernandez. Cooperative path-planning for autonomous vehicles using dynamic programming. In *Proceedings of the 2002 IFAC World Congress*, July 2002.

[4] B.O. Koopman. *Search and Screening: General principles with Historical Application*. Pergarnon, New York, 1980.

[5] M. Pachter and P. Chandler. Challenges of autonomous control. *IEEE Control Systems Magazine*, pages 92–97, April 1998.

[6] S.D. Patek, D.A. Logan, and D.A. Castanon. Approximate dynamic programming for the solution of multiplatform path planning problems. In *IEEE International Conference on Systems, Man, and Cybernetics, Conference Proceedings. 1999*, volume 1, pages 1061–1066, 1999.

[7] M. Polycarpou, Y. Yang, and K. Passino. A cooperative search framework for distributed agents. In *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pages 1–6, 2001.