Ashdin Publishing
International Journal of Swarm Intelligence and Evolutionary Computation
Vol. 2 (2013), Article ID 235569, 8 pages
doi:10.4303/ijsiec/235569

Research Article

Swarm Behavioral Inversion for Undirected Underwater Search

Albert R. Yu,[1] Benjamin B. Thompson,[2] and Robert J. Marks II[3]

[1]*Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA*
[2]*Applied Research Laboratory, The Pennsylvania State University, State College, PA 16804, USA*
[3]*Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798, USA*
Address correspondence to Robert J. Marks II, robert_marks@baylor.edu

Received 4 April 2012; Revised 5 February 2013; Accepted 13 February 2013

Copyright © 2013 Albert R. Yu et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Coordinating the dynamics of large groups, or swarms, of autonomous underwater vehicles in order to search a given target area can be difficult due to the plurality of the system, environmental complications, and the prolonged and indefinite duration of the patrol. This paper examines the use of swarm inversion to optimize the behavioral dynamics of a swarm of autonomous agents in a patrol search with underwater morphological and environmental constraints. In particular, the range of the forward-looking sensor range of agents varies spatially, requiring more search time in dark areas to maintain a high level of surveillance. This results in a tradeoff between the uniform coverage and surveillance frequency. The patrol fitness is determined via simulation feedback, and particle swarm optimization is used to invert and refine the behavior of the swarm. The tradeoffs between high search frequency and search uniformity are examined, as well as the evolved swarm's adaptability to varying environmental conditions and robustness of agent numbers. Results demonstrate that swarm inversion can yield effective agent behaviors for maintaining a presence in a given target zone despite stochastic navigation and an anisotropic environment.

Keywords swarm intelligence; multi-agent systems; patrol tactics; particle swarm optimization

1 Introduction

Unmanned vehicle autonomy is an increasingly active area of artificial intelligence research that seeks to implement effective decision-making algorithms in undirected vehicles. Designing the appropriate control scheme to achieve the desired vehicular response in all possible circumstances, and thus fully characterizing the explanation facility, is often a difficult if not unobtainable objective. Thus, the state-of-the-art autonomous control is largely comprised of ad hoc expert systems that may exhibit undesirable emergent behaviors when the operational theater is perturbed. Nevertheless, autonomous systems are necessary as direct and timely human control of all factors may be outside the capability of the operator, and deterministic control is limited by scope and adaptability, and thus has potential for vulnerability.

The details of autonomy become increasingly complex when dealing with unmanned, multi-agent systems. Large groups of autonomous, interacting agents, or swarms, can have emergent behaviors that are difficult to predict without simulation or physical implementation. However, the endeavor is worthwhile as large, interacting groups can often accomplish tasks individuals cannot, but the nature of mission execution and other subsequent emergent characteristics can be difficult to predict via analytic inspection [3, 10]. Swarm technology and evolutionary techniques address this issue by offering a robust and adaptive approach.

Swarm intelligence refers to large groups of agents interacting under simple rules that exhibit some emergent behavior and has applications in communications [5, 12], robotics [2], and optimization [6,17,18]. Prospective advantages of swarm intelligence include swarm robustness, plasticity, and decentralization [3], ideal characteristics for governing the interactions of a large group of autonomous vehicles. Emergent behaviors are often observed as a consequence of a given set of antecedents (e.g., sensor readings). The inversion of this process is to define criteria for the consequent or agent action and then refine the antecedents. Swarm inversion is the specific application of an optimization technique to multi-agent systems that seeks to develop optimal rules of operation for all agents by refining their behavioral responses. These responses are the focus of the optimization, since swarms that grow in capability can make the problem trivial. Variant techniques based on evolutionary algorithms have been applied to large groups, primarily in simulation and robotics [9,11,15,17,19].

Our proposed application of swarm inversion addresses the problem of dynamic undirected searches, specifically applied to an underwater patrol scenario. Here, a swarm of autonomous underwater vehicles with multi-channel sensing capabilities is given a limited amount of time to establish and maintain a presence in a given target zone. The primary difference between this scenario and similar work [8,9,11] is the inversion algorithm, the nature of the agent's control parameters, and specific underwater

morphological constraints. There is no external performance behavior being sought. Any emergent behavior that performs well within the imposed physics-based constraints is acceptable. We use the Combs swarm inversion method which involves the coevolution of disjunctive fuzzy logic [7]. The reader is referred to this work for a discussion of details on the inversion process as well as a discussion of prior work which leads to use of the Combs method.

The agents we use do not leave pheromone trails [8] for other agents to find nor will they follow waypoints or landmarks. They will not be able to directly communicate among each other or to any central controller. Although this agent ability could be added, neither of these capabilities were deemed characteristic of the agents we have in mind for the operation of surveillance [4].

An illustrative analogy to our problem is the patrol of a field at night by agents with limited vision. The field in some areas is illuminated so that the surveillance can be performed quickly over a large area. In darker areas, more time is required to assess safety status with the same certainty. The field can be occupied at a point at any time by an enemy agent, so the same regions must be repeatedly inspected. The average overall surveillance frequency of the field can be made high by agents spending all of their time patrolling well-lit areas. This, however, leaves darker areas uninspected for long periods of time. Requiring inspection of darker areas when there are fixed resources leaves well-lit areas less frequently visited. There is therefore a tradeoff between overall average surveillance frequency and the uniformity of coverage.

There are sensible, deterministic tactics to search a given terrain. Agents can line abreast and move in formation, comb the area, or follow a pre-planned path. However, path planning in an anisotropic environment is not a trivial task. Planned paths also display behaviors that are relatively easy to observe, ascertain, and subsequently circumvent. The stochastic nature of swarms makes the patrolling agents much more difficult to predict and counter, while the robust and adaptive nature of swarm intelligence would be advantageous in execution.

2 Model description

2.1 Patrol scenario

The underwater environment affects an agent's ability to search its surroundings by restricting communication and obscuring visibility. Unlike surface or aerial vehicles, an underwater vehicle utilizing acoustic sensing has limited channels available and often has few forms of direct communication. Thus, their interactions are modeled here as indirect and passive; agents become aware of each other by observing proximity noise or crosstalk. The underwater environment can also contain acoustic shadow zones, areas where deviations in the sound speed profile cause refraction in acoustic transmissions, limiting an agent's effective viewable distance. For our simulation, a high-level surface attenuation map is assumed to be known or approximately calculable by the agent, whether a priori or in real time via environmental readings.

2.2 Agent morphology and coverage maps

The swarming model considered assumes a high-level environment attenuation map. Agents are modeled to have a maximum speed and yaw rate, and their acoustic sensing capabilities are approximated as a visibility arc representing the ensonified area with the highest probability of detection by that agent. As an agent travels, the previously ensonified areas are retained as a tail representing a memory component that is only known to that particular agent (Figure 1). Each tail decays exponentially and eventually requires the agent to revisit and refresh these areas.

As each agent travels, an aggregate pixel coverage map is assembled, representing the combined coverage that a given pixel has been searched by any agent. This aggregate includes the decaying memory component of each agent. After a fixed iteration interval, the scenario is terminated, and the theater's final combined mean coverage and pixel standard deviation is recorded.

2.3 Visibility attenuation and interference

A high-level attenuation map is applied to the field. Each pixel in the map is assigned a value from 0 to 1 that represents a scale modifier to the agent's visibility. Lower values reduce the ensonified area of any agent on that pixel. Agents may also interfere with each other due to channel constraints. Whereas two agents in different channels will see each other if encountered, two agents operating in the same band generate crosstalk and confusion. Similarly, two agents within a close distance will generate proximity noise and overload all other acoustic signals, confusing both agents. This results in the agent's ensonified arc becoming void for that particular time step, and no contribution is made to the aggregate confidence map.

3 Swarm inversion

3.1 Genomic parameterization

The evolved agent genome is an array representation of each behavioral response parameter to a given sensor. A total of 10 evolvable parameters (initialized as uniform random variables over their entire dynamic range), characterize three primary sensors. Agents have sensors for their current position in the attenuation map via a Global Positioning System (GPS) or Inertial Navigation System (INS). Agents also have sensors for interference and are aware of the general direction but not range of the offending source. Finally, agents develop a response to the closest visible agent. Disjunctive fuzzy Combs control is achieved through

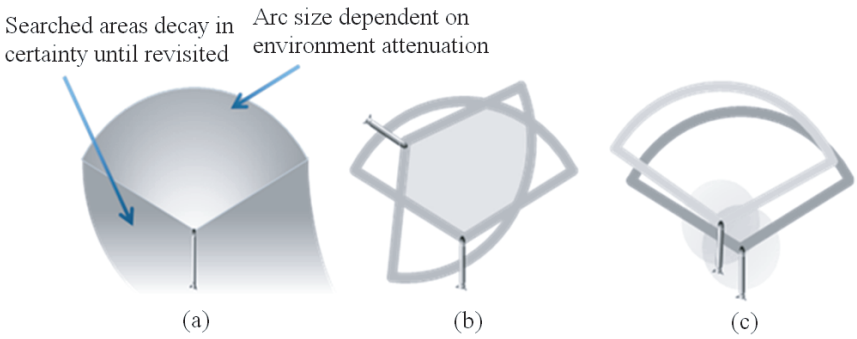

Figure 1: (a) Ensonified arc approximation with memory decay component, (b) channel interference, and (c) proximity interference: agents can conflict with each other in the above manners, and their resulting ensonified swaths are considered void for the relevant time steps. Agents indirectly communicate their relative bearing through this interference.

use of an activator function for each sensor. The activator function is parametered by a fuzzification vector. Each sensor consequent is scaled against an inertial component, aggregated, and applied as a final heading change decision subject to noise, yaw, and speed constraints. These three sensors and 10 evolvable parameters are:

(1) a_0, weighted response of vector toward nearest visible ally, if applicable;
(2) b_0, weighted response to direction of source of interference, if applicable; and
(3) $\{v_0, \ldots, v_7\}$, piecewise-linear visibility response to the attenuation map.

For sensors 1 and 2, the agent responds with a unit vector in the direction of the nearest visible ally or noise source scaled by the evolved parameter value. For sensor 3, each agent retains its previous two headings and visibility levels in order to estimate the local visibility gradient of the attenuation map. The agent then responds with a unit vector in the direction of the maximum decreasing gradient, multiplied by a factor determined by the piecewise linear function and its current visibility level. The maximum decreasing gradient is calculated from the cross-product of the two previous headings, adjusted for the agent's turn direction. This allows the agent to estimate the direction of decreasing visibility.

3.2 Fitness function

Developing a well-tuned fitness function is imperative to optimizing the swarm's behavior for this simulation. Gaudiano et al. [8] examined evolving state transition parameters for a multi-agent system of missiles, concluding that the inversion process's performance was heavily influenced by agent initialization and fitness function, and that the formulation of the fitness function could introduce unwanted biases. Small adjustments made to the fitness function can drastically shift the inversion's solution, and each solution may have a range of fitness values due to initializations and noise. Several known strategies in developing the fitness function include the use of prior knowledge to limit the search space and fixed or de-randomized initializations [12]. To reduce the impact of initialization on the performance of the swarm, agents are initialized randomly around a fixed ring at the center of the field and given an outward initial trajectory.

The goal of this inversion is to direct agents into searching the field frequently and uniformly. However, there is an inherent imbalance between these two factors; perfect mean coverage is unobtainable due to the limited number of agents, but perfect uniformity can easily be achieved if all agents interfere with each other, contributing nothing to the aggregate map and resulting in zero standard deviation. As this solution is relatively simple to discover, a third term regarding agent interference was required to prevent the evolution from circumventing the true goal of the mission. To this end, the three major objectives were to maximize mean coverage μ of all pixels in the zone, maximize uniformity of coverage via minimizing the standard deviation σ among all pixels on the map, and minimize average ratio of time spent blinded by interference b. A uniformity weighting factor λ was incorporated to tweak the fitness function and direct the optimization between mean and uniformity, expressed below. The λ variable is a tactical variable chosen by the user in accordance with the degree of importance of σ. In Pareto optimization, such parameters are commonly used to tune between competing attributes in the design process.

The goal is to maximize this fitness value through simulation feedback. An exponential term was used to reshape the fitness landscape to reward higher scores.

$$f_\lambda(\mu, \sigma, b) = e^{\mu - \lambda\sigma - b}. \quad (1)$$

3.3 Parameter inversion

Under default conditions with zero behavioral responses to the environment and ally interactions, agents produce a mean, per-pixel confidence map that reflects the high-level attenuation map, as depicted in Figure 2. The shadow zone is

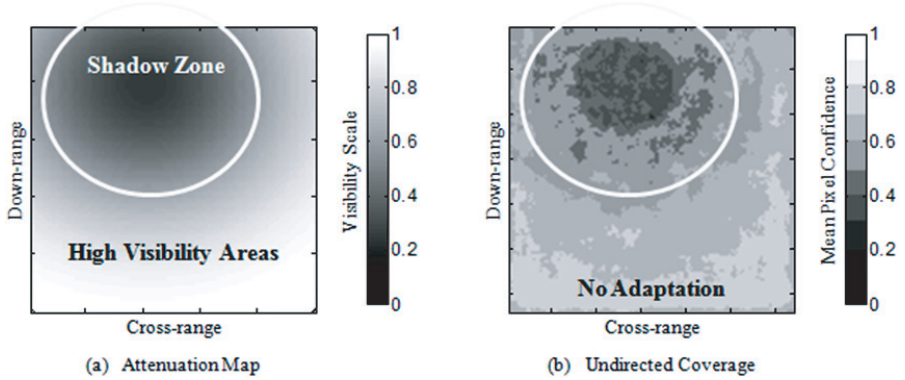

Figure 2: An (a) example high-level attenuation map representing a scaling modifier on an agent's effective visibility range and (b) the resulting mean pixel confidence of agents patrolling for 10,000 time steps with no specific responses to sensory inputs, displayed in contour form for clarity. Agents within the shadow zone in (a) have their visibility significantly reduced, and this is reflected in the mean patrol coverage (b) of the pre-evolved agents.

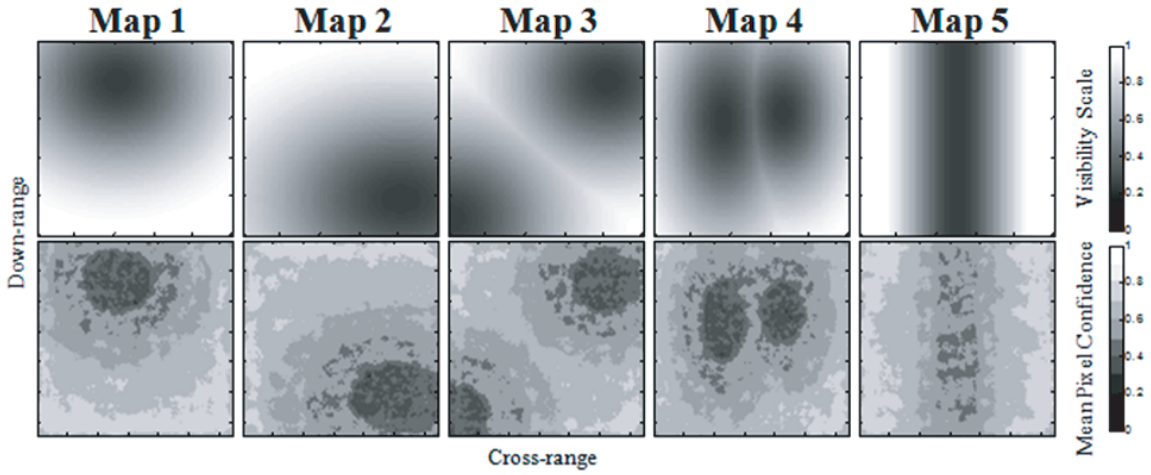

Figure 3: Swarms were evolved to search up to five different visibility attenuation maps, depicted in the upper row. With no specific behavioral responses, the swarms produce non-adapted mean pixel confidence maps (lower row) that reflect its respective attenuation map. The ideal swarm should search these areas uniformly at high mean confidence.

covered poorly on average relative to higher-visibility areas. An ideal swarm model should search all areas frequently and uniformly.

A variant of Shi and Eberhart's modified particle swarm optimizer (PSO) [20] with re-initialization is utilized in optimizing the agents' response functions. Each PSO agent is a solution genome that is run through the simulation in order to evolve the fitness of the population. In general, a population size of 100 genomes searching over 200 generations is used. The PSO is used to optimize the 10 evolvable parameters of the agent behavioral response genome.

3.4 Simulation setup and scenarios

For this simulation, a fixed number of homogeneous agents are initialized randomly about a ring formation at the center of a 128×128 pixel square theater. Agents may freely leave the field but are attracted to the center once outside theater bounds. The simulation's frame rate is fixed and corresponds to a maximum step size of 1 pixel, giving an effective maximum velocity of 1 pixel per unit time. All agents are synchronized, updating their actions simultaneously, once per frame or time step. Agents have a maximum viewable distance of 5 pixels and a memory decay rate of 0.99 per time step. The swarm is allotted 10,000 time steps to complete their patrol. In the base scenario, there are 60 agents limited to two channels. A C++ program was written to perform the simulation and execute the PSO.

Several simulation variants are tested. First, agents evolve on a fixed map (map 1 in Figure 3) with varying values of λ in order to observe the impact of the fitness function on average confidence and coverage uniformity. The second variant introduces the different attenuation maps

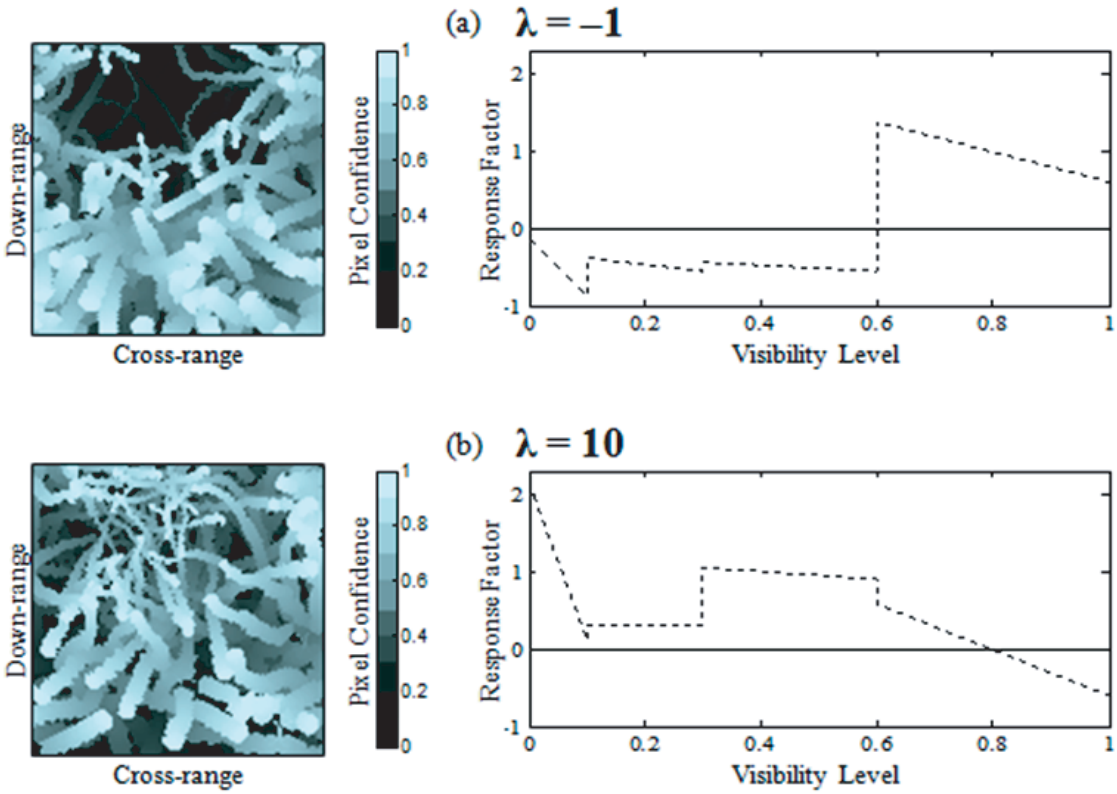

Figure 4: On map 1, the (a) the evolved swarm for $\lambda = -1$, emphasizing high coverage, and (b) the evolved swarm for $\lambda = 10$, emphasizing uniformity. Swarm (a)'s piecewise visibility response indicating repulsion from the shadow zone below 0.6 visibility scale. Conversely, swarm (b)'s response is mostly positive, with various levels of attraction toward the shadow zone.

in Figure 3 into the training process in order to improve universal performance and demonstrate swarm adaptability. Finally, swarm robustness is examined by observing the performance of the evolved swarms with varying numbers of available starting agents.

For the second variant, there is an issue with calculating a fitness function for different visibility maps. Fitness scores are not even across maps: some maps inherently have higher mean visibilities or standard deviations, which directly influences the swarm's performance and fitness calculation. To address this issue, quick optimizations are run separately for $\lambda = -1$ and $\lambda = 10$ on each map. The fitness values for these two results are mapped to zero and one linearly. A final evolution cycling through all five maps is used to generate the solution.

4 Results

4.1 Single map evolution using multi-objective fitness

Multi-objective optimization [14] demonstrates the inversion's ability to search the shadow zone given a specific fitness function. The fitness function portrayed in (1) yields a higher fitness value for swarms that achieve high mean coverage μ, low pixel image standard deviation σ, and low average blind-time ratio b. The weighting factor λ directly influences the fitness calculation: high values of λ correspond to a higher weighting on uniformity, or low standard deviation, while low values of λ signify higher emphasis on the overall mean pixel search. Intuitively, this means that when λ is low, the agents should avoid the shadow zone as it will decrease their visibility and thus the total mean pixel coverage. When λ is high, total coverage is deemphasized, and the evolution trades higher pixel coverage mean for the improved uniformity gained by searching the shadow zone.

A qualitative examination of the behaviors of the evolved swarms demonstrates these characteristics. Figure 2(a) presents the tested attenuation map. There is a readily apparent repulsion from the shadow zone in the $\lambda = -1$ solution, demonstrated in Figure 4(a). Most agents actively turn away from the shadow zone when they encounter the 0.6 visibility threshold. These actions reflect the largely negative repulsion in the evolved genome depicted in the simulation snapshot. For $\lambda = 10$, there is a visible swarming of the shadow zone due to the various levels of attraction provided by the corresponding piecewise response genome in Figure 4(b).

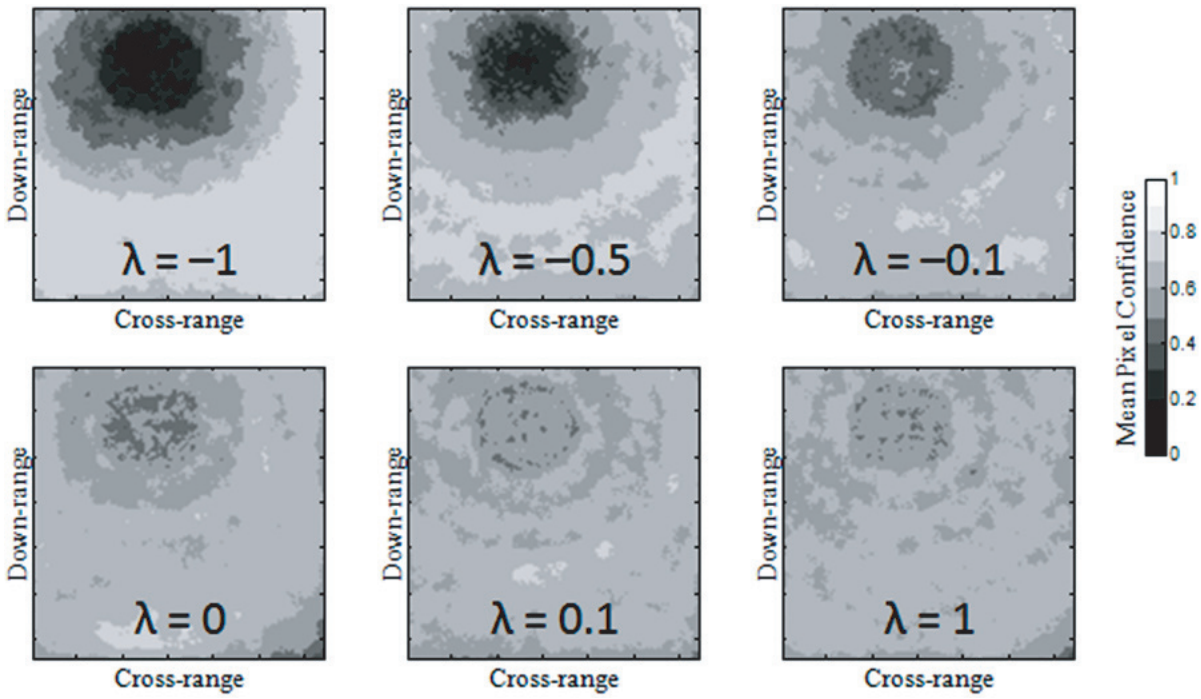

Figure 5: Final evolved confidence map examples for varying values λ. As λ increases, agents venture into the shadow zone as the fitness function rewards greater uniformity. There is a visible overall loss in mean coverage as more uniform solutions trade both the dark and bright gray zones for midrange values.

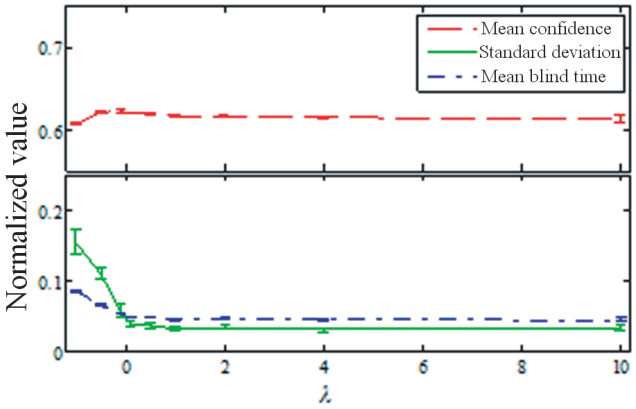

Figure 6: Map pixel mean confidence, standard deviation, and blind time for $\lambda = -1$ to 10, with 30 trials each. There is very little variation in mean and blind time beyond $\lambda = 1$.

Figure 5 depicts examples of the final resulting coverage maps for various values of λ and the transition from avoiding to searching the shadow zone. Due to random initializations and agent noise during the mission, final evolved confidence maps can vary noticeably in appearance. The results of 30 random initializations of the simulation for each value of λ are depicted in Figure 6. Despite the stochastic nature of the swarm leading to variations in final confidence maps for repeated runs, the final confidence maps are uniform. The evolution stresses lower standard deviation with increasing λ. To drive the swarm toward more uniform searches in subsequent calculations, a weighting factor of $\lambda = 10$ is used in the optimization process.

As expected, single-objective variants of the fitness function did not yield promising results. Simply maximizing the mean is insufficient as this encourages the agents to confine their search to areas of high returns, leading to agents avoiding the shadow zone. Alternatively, maximizing the minimum pixel confidence had trivial improvement over the same evolution time due to the strictness of the condition. Uniformity constraints were found to require the blindness term b as otherwise fitness was driven to zero via interference at the expense of high coverage.

4.2 Map training and adaptability

The performance of the swarm was dependent on the visibility map. Agents that were optimized for one attenuation map did not necessarily maintain their performance for alternative fields, as demonstrated in Figure 7. This was expected as the crafted fitness function and resulting evolutionary process was not map invariant. However, these evolutions were still useful, as they provided information on what range of μ, σ, and b an optimized swarm on a given map will yield. Various representative maps were needed in the training process in order to address adaptability, and these extremes provide a method for comparison.

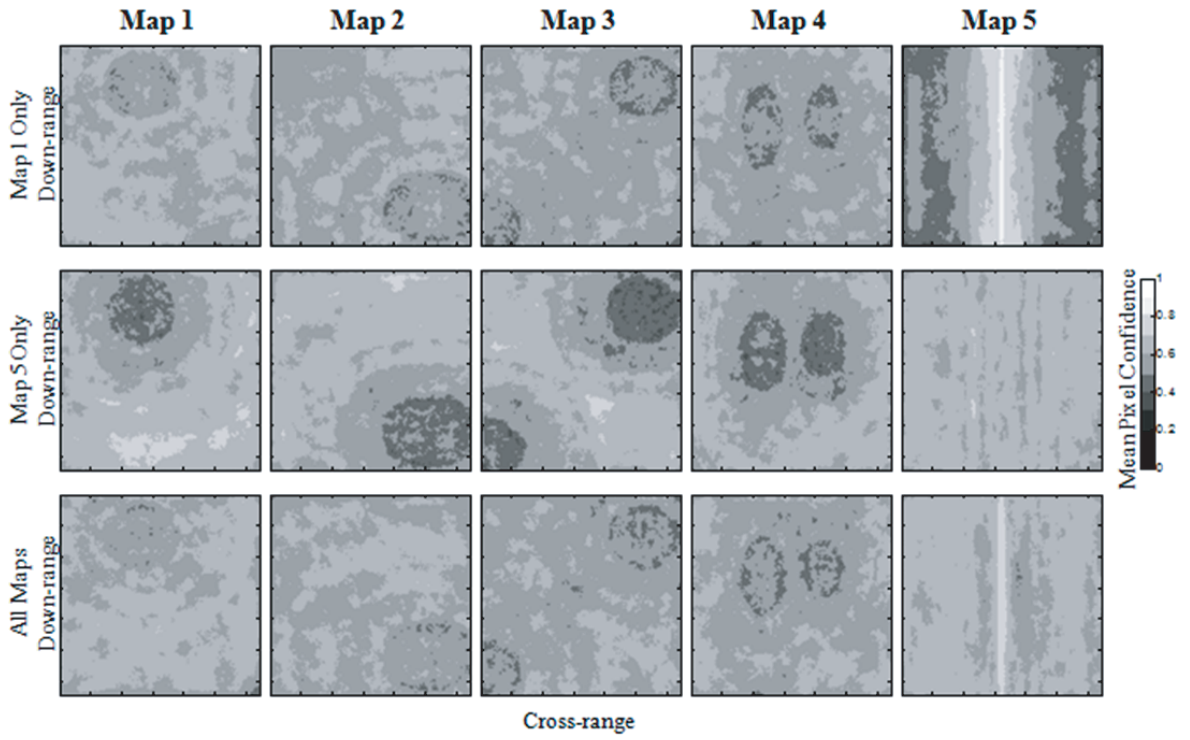

Figure 7: The performance of swarm trained specifically for map 1 (top row) or map 5 (middle row) suffers on alternate maps. The performance of the evolved solution trained on all five maps (bottom row), using normalized scores to calculate fitness. The performance of this swarm is comparable with that of solutions evolved specifically for each map.

Table 1: Average fitness scores on attenuation maps.

	Average fitness (30 trials each, $\lambda = 1$)				
Solution	Map 1	Map 2	Map 3	Map 4	Map 5
Map 1 only	1.702	1.678	1.649	1.648	1.516
Map 2 only	1.700	1.684	1.652	1.646	1.548
Map 3 only	1.693	1.675	1.651	1.642	1.502
Map 4 only	1.703	1.684	**1.653**	**1.655**	1.670
Map 5 only	1.656	1.634	1.601	1.617	**1.744**
Sum fitness	1.693	1.679	1.646	1.644	1.702
Normalized fitness	**1.705**	**1.685**	**1.653**	1.652	1.715

Simply summing the scores on all five training maps in order to calculate a genome's fitness is insufficient. Map scores vary with structure, leading to some maps rewarding disproportionately or having relatively lenient solutions. Map 5, in particular, improves the most, leading to the agents preferentially optimizing this map, often at the expense of the others. Normalizing the individual evolved performances was observed to reduce the bias in this process, displayed in Figure 7 (bottom row). Table 1 lists the average fitness values of 30 trials for each of the conditions. Agents trained on all the available maps with their fitness scores normalized performed consistently well on all maps. While map-specific evolutions often achieved the highest scores of any swarm for that map, they regularly underperformed on other attenuation maps.

4.3 Agent robustness

The robustness of the agents about the $\lambda = 10$ solution is depicted in Figure 8, calculated from 30 trials each. In the vicinity of 60 starting agents, there was little variation in the standard deviation of the agents. The mean confidence and average blind time do drift upward as agent numbers increase, but this is expected as more agents mean more coverage and also more opportunities for interference. However, these values do not change much. For the given circumstances, the swarm is robust and can maintain its performance despite slight variations in agent numbers.

5 Conclusions

Swarm inversion can be an effective tool in refining the behavior of a homogenous group of autonomous agents in order to complete a given task, often producing clever or unexpected solutions for the problem scenario. The classical advantages of swarms are demonstrable as the resulting agents were robust to changes in initial swarm size and adaptive to changes in the attenuation map. The inversion process was capable of developing an effective and robust behavioral guide for searching the given target zone.

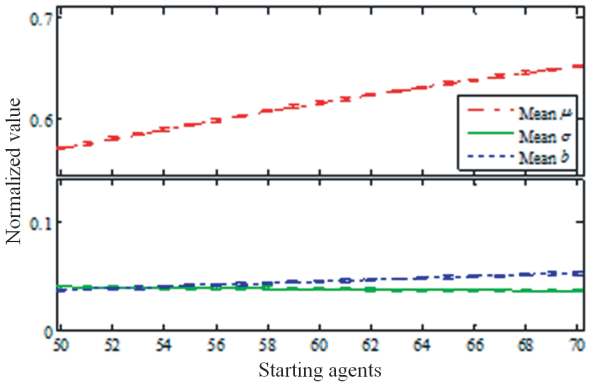

Figure 8: The robustness on map 1 with $\lambda = 10$ about 60 starting agents, taken from 30 trials each. Standard deviation is consistent, while mean coverage and blind time increase with the number of agents. This is expected, as more agents mean more areas searched as well as more opportunities to encounter other agents and cause interference.

The effect of the weighting factor in the fitness function has an appreciable effect on the evolved performance of the swarm, where increasing the value of λ increasing the relative importance of achieving higher uniformity. Training on a wide range of maps can improve the general operation of the swarm at the cost of individual map performance.

Further details on the swarms and simulations, including results and videos, are available online at www.neoswarm.com [1, 12, 13, 16, 21].

Acknowledgment This work was supported by ONR.

References

[1] P. Arabshahi, A. Gray, I. Kassabalidis, A. Das, S. Narayanan, M. El-Sharkawi, et al., *Adaptive routing in wireless communication networks using swarm intelligence*, in Proc. of the 19th International Communications Satellite Systems Conference, Toulouse, France, 2001.

[2] R. Beckers, O. E. Holland, and J. L. Deneubourg, *From local actions to global tasks: Stigmergy and collective robotics*, in Artificial Life IV, R. Brooks and P. Maes, eds., MIT Press, Cambridge, MA, 1994, 181–189.

[3] E. Bonabeau and C. Meyer, *Swarm intelligence: A whole new way to think about business*, Harvard Business Review, 79 (2001), 106–114.

[4] L.-D. Bui and Y.-G. Kim, *An obstacle-avoidance technique for autonomous underwater vehicles based on BK-products of fuzzy relation*, Fuzzy Sets and Systems, 157 (2006), 560–577.

[5] G. Di Caro and M. Dorigo, *AntNet: distributed stigmergetic control for communications networks*, Journal of Artificial Intelligence Research, 9 (1998), 317–365.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, *Ant system: optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26 (1996), 29–41.

[7] W. Ewert, R. J. Marks II, B. B. Thompson, and A. Yu, *Evolutionary inversion of swarm emergence using disjunctive combs control*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2013), 1–14.

[8] P. Gaudiano, E. Bonabeau, and B. Shargel, *Evolving behaviors for a swarm of unmanned air vehicles*, in Proc. of IEEE Swarm Intelligence Symposium (SIS 2005), Pasadena, CA, 2005, 317–324.

[9] J. Golbeck, *Evolving optimal parameters for swarm control*, in NASA/DoD Conference on Evolvable Hardware, Alexandria, VA, 2002, 152–153.

[10] I. A. Gravagne and R. J. Marks II, *Emergent behaviors of protector, refugee and aggressor swarm*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37 (2007), 471–476.

[11] T. Hussain, D. Montana, and G. Vidaver, *Evolution-based deliberative planning for cooperative unmanned ground vehicles in a dynamic environment*, in Genetic and Evolutionary Computation (GECCO 2004), vol. 3103 of Lecture Notes in Computer Science, Springer-Verlag, New York, 2004, 1017–1029.

[12] I. Kassabalidis, M. El-Sharkawi, R. J. Marks II, P. Arabshahi, and A. Gray, *Adaptive-SDR: adaptive swarm-based distributed routing*, in Proc. of the International Joint Conference on Neural Networks (IJCNN '02), Honolulu, HI, 2002, 351–354.

[13] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks II, P. Arabshahi, and A. A. Gray, *Swarm intelligence for routing in communication networks*, in IEEE Global Telecommunications Conference, San Antonio, TX, 2001, 3613–3617.

[14] J. Knowles, *Closed-loop evolutionary multiobjective optimization*, IEEE Computational Intelligence Magazine, 4 (2009), 77–91.

[15] H. Kwong and C. Jacob, *Evolutionary exploration of dynamic swarm behaviour*, in The 2003 Congress on Evolutionary Computation (CEC '03), Canberra, Australia, 2003, 367–374.

[16] NeoSwarm.com, *http://www.neoswarm.com/zamboni.html*.

[17] H. Okada and T. Takagi, *Evaluation of multi-objective genetic algorithm for RoboCupSoccer team evolution*, in SICE Annual Conference, Tokyo, Japan, 2008, 151–154.

[18] R. D. Reed and R. J. Marks II, *An evolutionary algorithm for function inversion and boundary marking*, in Proc. of IEEE International Conference on Evolutionary Computation, Perth, WA, 1995, 794–797.

[19] C. W. Reynolds, *Evolution of corridor following behavior in a noisy world*, in From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior, D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, eds., MIT Press, Cambridge, MA, 1994, 402–410.

[20] Y. Shi and R. Eberhart, *A modified particle swarm optimizer*, in IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, Anchorage, AK, 1998, 69–73.

[21] A. R. Yu, *Optimizing multi-agent dynamics for underwater tactical applications*, master's thesis, Department of Electrical and Computer Engineering, Baylor University, Waco, TX, 2011.