

# Evolutionary Inversion of Swarm Emergence Using Disjunctive Combs Control

Winston Ewert, Robert J. Marks, II, *Fellow, IEEE*, Benjamin B. Thompson, and Albert Yu

**Abstract**—Given simple agent rules, a swarm’s emergent behavior can be difficult to predict. The inverse problem is even more difficult: Given a desired emergent behavior, what are the rules by which swarm agents should operate? Disjunctive fuzzy control is proposed as a method to model swarm agents. Compared to more commonly used conjunctive fuzzy control such as that proposed by Mamdani, disjunctive fuzzy control is robustly fault tolerant and disjointly connected. Swarms are inherently disjunctive. Instead of agents working in coordination with one another, each swarm agent contributes individually to the result. The disjunctive attribute can also be applied at the sensor level for each individual agent. Disjunctive control allows adaptation of the describing membership function, as is commonly done in conjunctive control. The inversion process is illustrated with numerous simulation examples, including a predator–prey game, gang warfare, and escaping agents. The swarm is instructed what to do but not how to do it. Imposition of fitness constraints and repeated generations of evolutionary molding of agent performance can then result in unexpected emergent behaviors of the swarm, e.g., use of decoys, self-sacrifice, flanking maneuvers, and shielding of the weak.

**Index Terms**—Disjunctive control, emergent behavior, fuzzy control, inverse problem, swarm intelligence.

## I. INTRODUCTION

SWARM INTELLIGENCE is based on the emergent behavior of groups of individual social agents performing simple tasks. Certain insects [16] and bacteria [86] are examples. Swarm intelligence has found application in telecommunications [9], [24], [25], [53], [54], business [17], [60], robotics [13], [15], [27], [43], and optimization [31], [32] and makes use of a plurality of highly disjoint agents interacting using simple rules. Simple swarm algorithms have been employed to assist with load balancing of peer-to-peer networks [67], routing within mobile *ad hoc* radio networks [40], self-organizing construction and assembly [58], and even health care [41], [91].

Simulations of swarm algorithms have parameters that can be tuned [98]. Pioneering works have focused on enhancement

Manuscript received April 12, 2012; revised July 3, 2012; accepted September 4, 2012. Date of publication February 1, 2013; date of current version August 14, 2013. This work was supported in part by the National Science Foundation under Grant ECCS 0801440 and in part by the Center for Evolutionary Informatics. This paper was recommended by Associate Editor S. Das.

W. Ewert and R. J. Marks II are with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798 USA (e-mail: winstonewert@gmail.com; robert\_marks@baylor.edu).

B. B. Thompson is with the Applied Research Laboratory, The Pennsylvania State University, State College, PA 16804 USA (e-mail: bbt10@arl.psu.edu).

A. Yu is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195-2500 USA (e-mail: alreyu@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2012.2227252

of the emergent behavior for which the swarm is designed [2], [4], [12], [14], [38], [44], [45], [71], [74]. We focus on a more general problem of evolving unspecified emergent behavior based on goal without regard to the manner in which success is achieved. Results are often unexpected. In one case, for example, sacrificial agents were evolved which act as decoys to distract predators with the goal of maximizing the lifetime of the swarm collective. In another, an evolved swarm developed deceptive flanking tactics to avoid capture. Some swarms maximize their life span by being aggressive toward the enemy. Other swarms extend their lifetime by prolonged strategic retreat.

Often, determination of emergent behavior from simple rules of interaction in swarm intelligence escapes both analytic and intuitive inspections [39]. Here are some examples from the literature.

- 1) Each agent randomly roams on a floor covered with wood particles, picking up a particle if it bumps into one. When an agent bumps into a second particle, it unloads its load. Now empty handed, it continues roaming looking for another particle, and the process is repeated [16].
- 2) Each agent randomly identifies two other agents and moves to place itself between the two agents [17].
- 3) Each agent randomly identifies two other agents and tries to move such that one agent, a protector, is between it and the other agent, an aggressor.

The rules in these are expressed clearly and without ambiguity. The identification of the emergent behavior of the swarm, however, is not readily evident.<sup>1</sup>

These three simple examples of the forward swarm problem illustrate the difficulty of the analysis of emergent behavior in even simple swarms.<sup>2</sup> The necessity of inclusion of stochastic chaotic behavior in swarm models further complicates intuitive analysis [16], [26]. The inversion of the swarm, or swarm design, is even more daunting. Given a desired emergent behavior, what is the set of simple rules needed? Using fuzzy Combs control, we investigate such a design applied to a plurality of cases. We also show that Combs control can, on occasion, allow identification of local agent performance properties that gives rise to evolved emergent behavior.

This paper has two distinct parts. The first provides a short review of disjunctive Combs control in comparison to the more widely used conjunctive Mamdani control. Combs control is

<sup>1</sup>Once an emergent behavior is identified, however, the relationship between the rules and the emergent behavior can become more clear.

<sup>2</sup>The emergent behaviors for these three cases are in the Appendix.

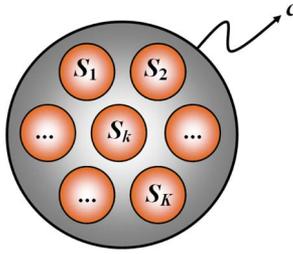


Fig. 1.  $p$ th agent in a swarm team of  $P$  agents. As shown here, each agent has  $K$  sensors  $\{S_k | 1 \leq k \leq K\}$ . The  $k$ th sensor makes a reading of  $s_k$  which is subjected to the actuator function in (4) to generate the sensor consequent  $c_k$ . The  $c_k$ 's from all sensors are aggregated to generate the scalar consequent  $c$  for the agent. When there is more than a single consequent, we denote them as  $c[1]$ ,  $c[2]$ , etc. In the examples in this paper, each agent has two consequents: one for each of two dimensions of movement. Each consequent can have a separate set of actuator functions.

shown to be more effective in the swarm inversion process largely because the corresponding search dimension [28]–[30], [79] is reduced. We demonstrate that Combs control is equivalent to the use of actuator functions. The manipulation of these actuator functions affects agent actions and, consequently, the emergent behavior of the swarm. The second part of this paper presents a number of swarms evolved using Combs control. Although the swarm evolution is seeking to increase an imposed fitness, the purpose of the simulations is not to maximize fitness or speed of convergence but, rather, to observe emergent behaviors in the swarm. These emergent behaviors are often unexpected yet are effective and, upon reflection, reasonable.

## II. SWARM FORMULATION AND CONTROL

We now make the case that evolutionary determination of disjunctive fuzzy logic parameters [22], [23], [94]–[97] is ideally suited for evolving the emergent behavior of swarms.

There are two levels of control in the swarm.

- 1) *Agent control.* There are  $P$  swarm agents on a team. One is shown in Fig. 1. Each agent has  $K$  sensors  $\{S_k | 1 \leq k \leq K\}$ . For our simulations, a homogeneous swarm is used so that each agent has the same resident control rules. Fuzzy logic is used to control each agent. Each sensor provides antecedents to the fuzzy control of each individual agent. A swarm of  $P$  agents is shown in Fig. 2.
- 2) *Global swarm control.* For the second level of the swarm control, the consequents of each agent action provide the emergent behavior  $\mathbb{C}$  of the swarm. The individual agent actions are aggregated into the overall swarm performance which is measured by a fitness function. For a swarm of prey, for example,  $\mathbb{C}$  might denote the median survival time of all agents, i.e., the time it takes for half of the swarm to be destroyed. In the common scenario of randomness within the swarm, including initialization and agent jitter,  $\mathbb{C}$  will be a random variable whose character can be assessed only through repeated trials [98].

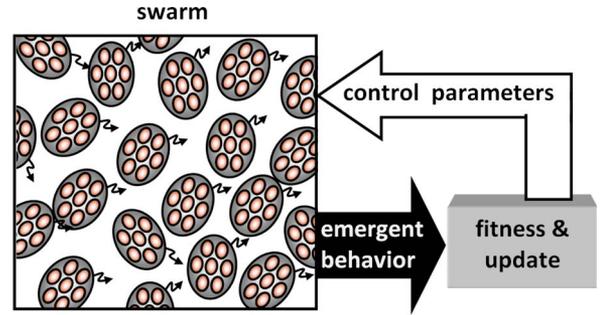


Fig. 2. Collection of  $P$  agents of the type shown in Fig. 1 form a swarm. We consider only the case where the fuzzy control parameters for all agents are identical. Each agent has a consequent in accordance to its sensor readings. Acting in concert, the swarm has a goal to achieve as measured by a fitness measurable only after the swarm operates. Inversion is formed using the standard search feedback shown. Through evolutionary inversion of actuator functions, the swarm can be made to perform better and better. Our simulations use particle swarm to update the control parameters. The manner in which the swarm achieves its global objective consequent ( $\mathbb{C}$ ) through emergent behavior can be unexpected.

## III. CONJUNCTIVE VERSUS DISJUNCTIVE FUZZY CONTROL

For each swarm agent consequent, traditional Mamdani conjunctive implication [63], [64], [66] can be expressed as

$$\bigcap_{k=1}^K A_k \rightarrow C \quad (1)$$

where  $A_k$  is the  $k$ th fuzzy descriptor of the fuzzification of the  $n$ th sensor and  $C$  is the fuzzy consequence.

For example, consider automobile control, and let  $C =$  “turn slightly right.” The Mamdani control rules then take on the following form:

$$\begin{aligned} \text{If } (A_1 = \text{turn front tires slightly right} \quad \text{AND} \\ A_2 = \text{turn rear tires slightly left} \quad \text{AND} \\ A_3 = \text{lightly brake right tires} \quad \text{AND} \\ A_4 = \text{lightly accelerate left tires}) \\ \text{THEN } C. \end{aligned}$$

Disjunctive implication used in Combs [21] control<sup>3</sup> is

$$\bigcup_{k,n} (A_k \rightarrow C). \quad (2)$$

Here, the contribution of each sensor to the agent’s performance is aggregated to assess the resultant consequent. The corresponding Combs control rules in the car turning example are

$$\begin{aligned} (\text{If } A_1 = \text{turn front tires slightly right} \quad \text{THEN } C) \quad \text{OR} \\ (\text{If } A_2 = \text{turn rear tires slightly left} \quad \text{THEN } C) \quad \text{OR} \\ (\text{If } A_3 = \text{lightly brake right tires} \quad \text{THEN } C) \quad \text{OR} \\ (\text{If } A_4 = \text{lightly accelerate left tires} \quad \text{THEN } C). \end{aligned}$$

<sup>3</sup>Application of disjunctive implication to fuzzy inference is commonly called the *Combs method* [36], [48], [50], [87].

In propositional Boolean logic,<sup>4</sup> there is an identity between the disjunctive and conjunctive implications dubbed the *law of importation* [50]

$$\left(\bigcap_{k=1}^K A_k \rightarrow C\right) \equiv \left(\bigcup_{k=1}^K (A_k \rightarrow C)\right). \quad (3)$$

The fuzzy logic generalization of the law of importation is not an identity. There is, however, often commensurate performance in comparison to the use of fuzzy conjunctive Mamdani rule matrices [50], [96].

Details contrasting the characteristics of conjunctive and disjunctive implications can be found elsewhere [22], [23], [94]–[97]. Combs control has many advantages. If sensors are lost, for example, an agent can straightforwardly adapt by seamlessly applying redundant resources. Likewise, new sensors can be easily added. The conjunctive form is brittle in comparison. Loss of a sensor requires reassessment of the implication (Mamdani rule matrix) structure.

If there are  $K$  sensors and a single consequent each requiring  $\{N_k | 1 \leq k \leq K\}$  fuzzy sets, the Mamdani fuzzy rule matrix for implementation requires

$$N_{\cap} = \prod_{k=1}^K N_k$$

fuzzy rules. If all of the  $N_k = N$  for all  $k$ , then  $N_{\cap} = N^K$ . Disjunctive control, on the other hand, requires

$$N_{\cup} = \sum_{k=1}^K N_k$$

rules [22], [23], [94]–[97]. If the number of fuzzy sets is the same, then  $N_{\cup} = NK$ . The number of fuzzy rules therefore increases linearly with respect to the number of antecedents rather than exponentially.<sup>5</sup>

In the inversion of the swarm, we will be searching through a space whose dimension is determined by the number of fuzzy rules. Therefore, besides its operational advantages, disjunctive control reduces the search space size, thereby avoiding the *curse of dimensionality* [79] for the swarm inversion process.

### A. Functional Representation

We now show that fuzzy Combs control can be reduced to the use of actuator functions for each sensor followed by aggregation.

An example of  $k$ th sensor contributes to the agent consequent  $A_k \rightarrow C$ . If the  $k$ th sensor is tessellated into  $N_k$  fuzzy membership functions, any scalar measurement, for example,  $s_k$ , will be fuzzified into a vector of  $N_k$  membership values. For disjunctive control, all of the elements in this vector are used to weight the fuzzy membership functions of the single consequent which is then defuzzified into a single crisp consequent scalar for the

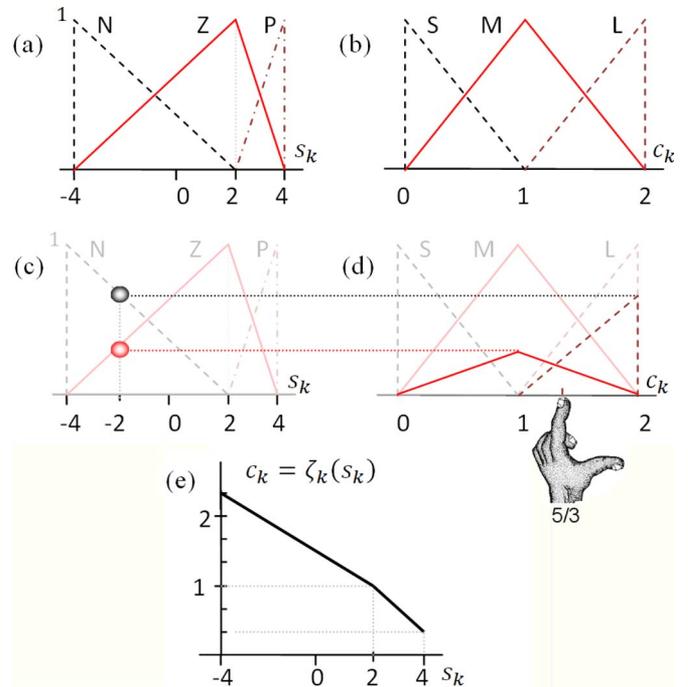


Fig. 3. Illustration that the actuator nonlinearity from disjunctive fuzzy rules is a 1-D equivalent to a fuzzy control surface. The antecedent and consequent fuzzy membership functions shown respectively in (a) and (b) follow the rules “ $N \rightarrow L$  AND  $Z \rightarrow M$  AND  $P \rightarrow S$ .” For a sensor reading of  $s_k = -2$ , the antecedent fuzzy membership functions for  $(N, Z, P)$  read  $(2/3, 1/3, 0)$ . This is illustrated in (c). To defuzzify, the consequent membership functions are weighted as shown in (d) and added. The center-of-mass balance point of the sum of the weighted consequent functions, as shown in (d), is  $c_k = 5/3$ . Thus,  $\zeta_k(-2) = 5/3$ . Repeating this process for all  $s_k \in [-4, 4]$  generates the piecewise linear actuator function  $\zeta_k$  shown in (e). To our knowledge, this is the first time that Combs control has been shown to equate to the use of an actuator function.

$k$ th sensor  $c_k$ . From this process, we conclude that each sensor reading is assigned a single consequent value which, in turn, is aggregated with the consequents of the other sensors to specify the composite consequent for the  $p$ th agent. Thus, *the value of the consequent is a simple 1-D function of the sensor antecedent* which we can write as

$$c_k = \zeta_k(s_k). \quad (4)$$

The *actuator functions*  $\zeta_k(\cdot)$  are the 1-D equivalent to the control surfaces in fuzzy inference systems [19], [49]. A detailed example is shown in Fig. 3.

Fuzzification and defuzzification of membership functions for disjunctive fuzzy control have been performed in previous treatments [22], [23], [94]–[97]. As shown in Fig. 3, however, formation and adaptation of actuator functions can be performed directly without consideration of the intermediate fuzzy components.

In summary, sensor  $S_k$  on an agent measures  $s_k$  from which we find the consequents  $c_k$  using the actuator function in (4). The  $c_k$ ’s are then combined using, for example, an aggregate function [18], into each agent’s consequent  $c$ .

Each agent may have more than one consequent in which case  $c_k[1]$ ,  $c_k[2]$ , etc., are generated using possibly different actuator functions acting on the same sensor inputs. Each is aggregated into consequents  $c[1]$ ,  $c[2]$ , etc., for each agent. The

<sup>4</sup>For implication,  $0 \rightarrow 0$  is 1,  $0 \rightarrow 1$  is 1,  $1 \rightarrow 0$  is 0, and  $1 \rightarrow 1$  is 1.

<sup>5</sup>A common observation is that conjunctive control with more rules and, therefore, more degrees of freedom has greater fine-grained flexibility than disjunctive control.

interaction of all agent actions then contributes to the emergent behavior of the swarm.

### B. Relative Performance

Using Combs control, the inverse problem [11], which has found use in many areas of computational intelligence [33], [46], [47], [51], [52], [80], [81], reduces to adapting the actuator functions to maximize the fitness of the emergent behavior  $C$ . This is a variation of fuzzy membership function adaptation applied in Mamdani-type inference systems [6]–[8], [65], [88] and similar to inversion of trained neural networks. Although Mamdani control requires exponentially more tunable parameters than Combs control, their performances have been shown to be commensurate in many applications [22], [23], [94]–[97]. Combs control then emerges in preference because of the stark reduction in the search space dimensionality.

There are other alternatives to Combs control other than adaptive Mamdani control, including neural networks [3], [79] and support vector machines [1], [42], [68]–[70], [82].<sup>6</sup> After training, both Combs fuzzy control and Mamdani fuzzy control offer a straightforward explanation facility for performance understanding. This is in contrast to black-box methods like neural networks where extraction of rules can be more difficult [81]. Explanation facilities are of specific interest in our work, since understanding the cause of unusual emergent behavior is important for the formulation of new tactics. We will show how evolved actuator functions can be examined to reveal swarm strategies. Arguably, Combs control offers a more transparent explanation facility than Mamdani control.

## IV. INVERSION EXAMPLES

Using Combs control of agents, here are some examples of emergent behaviors observed from inversion of competitive swarms. Results will be illustrated with figures, but videos of the swarm performance are more instructive. They are available online [72]. Each example has some variations as to engagement rules; however, the nature of the disjunctive control is the same in all of the cases. Elucidation of fine-grained details in each program is beyond the scope and length constraints of this paper. Details, though, can be found in documented code also available online [72]. All simulations are in Python and were run on a Linux PC cluster.

For the examples to follow, all swarm agents on a team contain the same sensors, actuator functions, and control rules. A performance fitness function for the swarm is chosen, and the swarm is run. Since there are stochastic components in every simulation (e.g., initialization and jitter), the fitness is a random variable, and each run of the simulation generates a single sample of the underlying random variable. Running a swarm from start to finish<sup>7</sup> results in a single stochastic fitness value. Using the measured fitness of a number of simula-

tions of swarms with different actuator functions, personal and global best results were noted, and after updating the actuator functions consistent with *particle swarm* search [34], [55]–[57], [84], [85], [92], another generation of swarm performance is assessed in another generation of swarms. Application of repeated generations resulted in emerged behavior that displays a high fitness. We refer to numerous evolutionary steps in the optimization of a swarm team as an *era*.

To address the stochastic contribution of swarm interaction, each swarm was run for 20 realizations, and the measured fitnesses of these swarms are averaged to estimate the mean of the underlying random variable. The particle swarm uses a population of 80 agents and was run for ten iterations at a time. The particle swarm algorithm has parameters CG and CP which weight the attractiveness of a particle to the global best of the swarm and the particle's personal best. [34], [55]. Both parameters were set to the commonly used default value of 2.0

All swarm agents exist on a 2-D square planar *playground* using floating-point coordinates between 1 and  $-1$  for both axes. All agents must remain within the playground at all times. Directions are kept on a trajectory by bounding increments in velocity change. At each time step, a velocity is added to the current position. Additionally, a small amount of random jitter [73], [76]–[78] is added to the current position. A number of consequents are assigned to each agent in a manner specific to each game.

For each sensor, an adaptive actuator function  $c_k = \zeta(a_k)$  is formed for each of the consequents. The shape of  $\zeta(a_k)$  is defined by three values of  $s$ , at the points 0, 1, and 2. The values are then connected in a piecewise linear fashion to form the actuator function. The actuator function has a range for  $c$  of  $[-1, 1]$ . The point locations are evolved to maximize the desired emergent behavior property. For the simulations to follow, the control properties of all of the agents on a team are the same.

### A. Bullies and Dweebs

The first game is based on a predator (bullies) and prey (dweebs) model. The dweebs are killed when they come into contact with the bullies. The bullies and dweebs begin uniformly randomly distributed across the playground. Only the dweeb strategy is evolved.

Both the bullies and dweebs make use of the same sensors as follows:

- 1)  $S_1$  = distance to the nearest team agent;
- 2)  $S_2$  = distance to the nearest agent on the other team;
- 3)  $S_3$  = distance to the center of the playground.

They both have the following consequents:

- 1) movement toward the nearest enemy agent;
- 2) movement toward the nearest friendly agent;
- 3) movement toward the center of the play area.

The swarm's fitness is defined as

$$C = \sum_{i=1}^T d_i \quad (5)$$

where  $d_i$  is the number of living dweebs at time step  $i$  and  $T = 2000$ . The dweebs are evolved around the fixed operation of the

<sup>6</sup>Interestingly, the training of such adaptive structures can, themselves, be controlled using fuzzy [5], [7], [20], [65], [89] or other [59], [75] control.

<sup>7</sup>In many instances, a stop criterion is imposed before the swarm runs its total course.

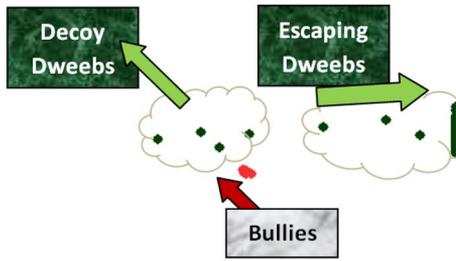


Fig. 4. Bullies are distracted by the decoy dweeb; eventually, a single sacrificial dweeb emerges and is chased by the bullies until it is caught. The remaining dweeb seek refuge as far away from the bullies as possible. The simulation software and a more illustrative and insightful video of the evolved swarm (video #1) are on NeoSwarm.com [72].

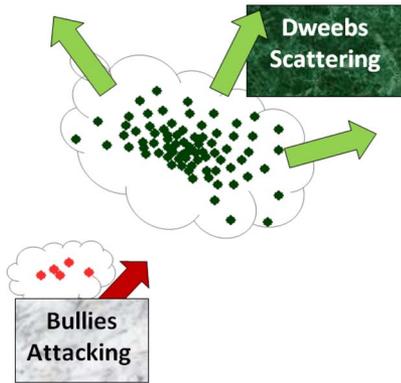


Fig. 5. Swarm of dweeb spreads out as the bullies attack. The simulation software and a more illustrative and insightful video of the evolved swarm (video #2) are on NeoSwarm.com [72].

bullies. Variations were observed to emerge dependent on the fixed point behavior of the bullies. Two cases are considered.

1) *Slow Chases*: In this scenario, the speed of a bully toward the nearest dweeb is set proportional to the separating distance, i.e., a bully will run quickly toward a dweeb that is far away and slowly toward a dweeb nearby. The dweeb are able to find a “sweet spot” where they can run slowly immediately in front the bully and avoid getting caught. One of the dweeb would typically “dance” with the bully while the others would hide out of range.

The interesting emergent dweeb behavior is one of self-sacrifice. One dweeb at a time attracts the bullies in a prolonged chase while the remaining dweeb move to a nonthreatening position. Eventually, the sacrificial dweeb is killed. After transient activity, a fresh self-sacrificing dweeb emerges, and the cycle continues. This emergent self-sacrifice strategy performed the desired function of maximizing the overall lifetime of the dweeb swarm as measured by the fitness. See Fig. 4 and the video available online [72].

2) *Center*: In the second scenario, bullies attack dweeb at a speed independent of their separation. Swarm inversion results in the dweeb strategy of clustering in the center of the playing field. The bullies attack, causing a scattering of the dweeb. The bullies then concentrate on sacrificial dweeb while other dweeb return to temporary safety in the center. As in the *slow chase*, the effect was of self-sacrifice albeit not as dramatic. See Fig. 5 and the video available online [72].

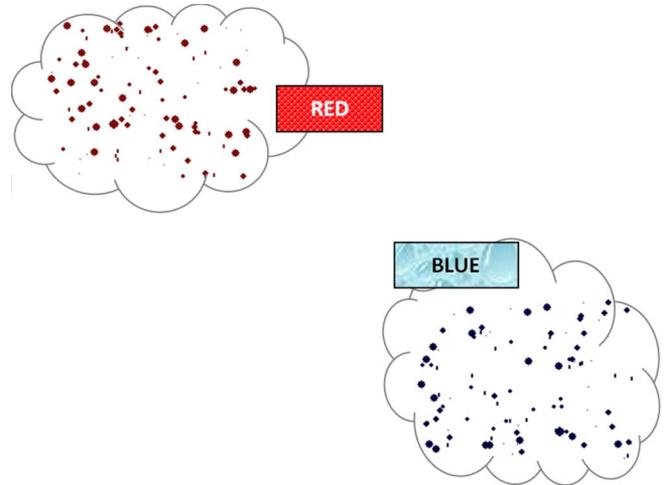


Fig. 6. Initial positions of the opposing gangs. Snapshots of the conflict are shown in Figs. 8 and 9. Also see the online video [72].

### B. Gang Warfare

In *gang warfare*, a second type of wargame, there are two gangs, red and blue, both able to retreat or to attack and kill the other. Each team agent has a randomly assigned strength, and when agents collide, the weak is killed and the strongest agent survives. However, the winning agent loses strength. When an agent is killed, the strength of the survivor is decayed by a factor of 0.9. Furthermore, when an agent is far enough away from any enemy agent, its strength slowly increases. If the nearest enemy agent is at least 0.1 units away, strength increases by 0.001 per iteration. Both teams are alternately evolved, and a number of different strategies emerge.

The sensors used for both the red and blue teams are as follows:

- 1)  $S_1$  = distances to the nearest enemy agent;
- 2)  $S_2$  = agent’s self-assessment of its strength;
- 3)  $S_3$  = strength of the nearest enemy agent.

The two consequents for each agent are as follows:

- 1) movement toward the nearest enemy agent;
- 2) movement perpendicular to the nearest enemy agent.

Fitness is defined to be

$$C = \sum_{i=1}^T (t_i - e_i) \tag{6}$$

where  $t_i$  is the number of agents of one team alive at time step  $i$  and  $e_i$  is the number of agents alive on the opposing team at time step  $i$  and  $T = 2000$ .

The gangs are alternately evolved. Blue is first. As shown in Fig. 6, the two gangs begin at the opposite sides of the play area. Snapshots of the game are shown in Figs. 6–9. Also see the video [72].

Different strategies are observed to arise.

#### 1) *Orbiting*

Strategically, it makes sense to attack the weak members of the enemy forces while keeping your own weak members out of the fight. The agents begin from the initial configuration in Fig. 6. The strong and weak blue agents

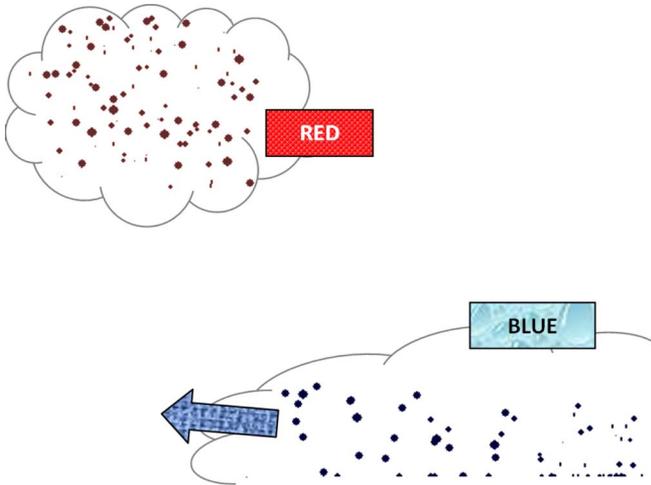


Fig. 7. Blue team divides.

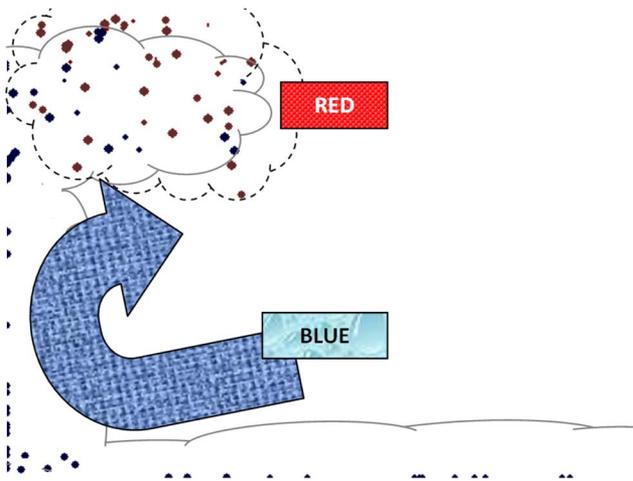


Fig. 8. Attacking from the west wall.

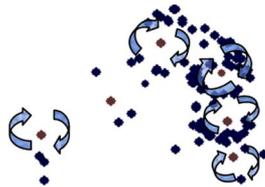


Fig. 9. Red clockwise circling the blue finishes the contest.

separate in Fig. 7. The stronger agents charge into the enemy territory to kill off the enemy agents in Fig. 8. After the weak members have been killed off, the agents spin around the stronger enemy agents in Fig. 9.

The orbiting action comes from the agents moving both toward and perpendicular to the enemy agent. Spinning

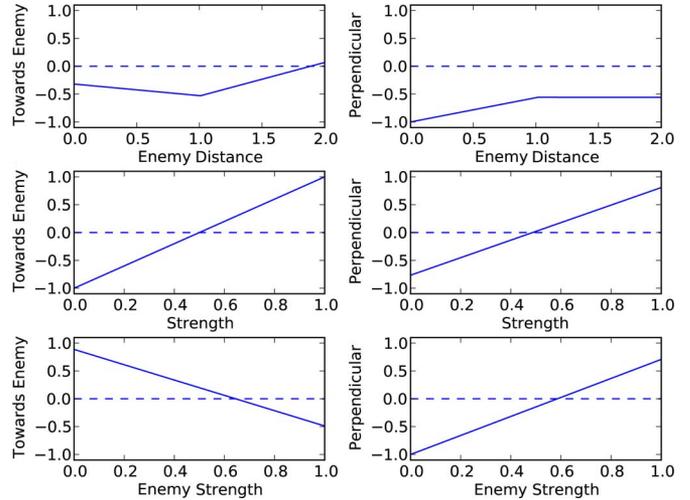


Fig. 10. Actuator functions evolved for the blue agents as used in Figs. 6–9. The upper left function indicates that the agent should retreat from the enemy unless the enemy is far away. The middle left function indicates that, the stronger the blue agent, the more that it is inclined to move toward its enemy. Conversely, as shown in the bottom left, the stronger an enemy, the more inclined an agent is to retreat. The clockwise circular motion of the blue agents around the red agents in Fig. 9 is due to the actuator functions controlling perpendicular motion. As is the case with satellites orbiting the Earth, the combination of perpendicular motion (the satellite’s momentum) and attraction to Earth results in orbiting.

around the agent would possibly bring a weaker agent into view. In the end stage, the perpendicular movement prevents the circling agents from coming into contact with a strong enemy agent and being killed.

Fig. 10 shows the actuator functions optimized for the blue to fight this battle.

2) *Defensive strategy*

After the blue team is evolved for an era, its behavior is fixed, and the red team is evolved to counter the performance of the blue team. In this section, we look at the effects of the blue team being evolved a second time to counter the counters of the red team. We denote this process by  $\rightarrow B \rightarrow R \rightarrow B$ . Each right arrow denotes an evolution era. Snapshots of the swarming are shown in Figs. 11–15. Also see the online video [72]. The blue team has evolved a defensive strategy. Instead of attempting to kill the enemy agents, the strategy now prioritizes not getting killed.

a) Fig. 11. The battle begins as shown. The blue team is applying a defensive strategy. The stronger blue agents move closer to the red enemy, thereby attracting their attention. The weaker blue agents retreat to the south-east corner.

b) Fig. 12. The red enemy agents continue to chase the blue agents but do not attack because the blue agents have too much strength. The weaker blue agents at the bottom right gain strength over time while this is happening.

c) Fig. 13. The blue agents in the southeast corner have gained strength and are moving up the east wall in order to join the battle. The blue agent cluster at the top of Fig. 12 has bifurcated. One portion is in the middle of a string of red agents who are circling in

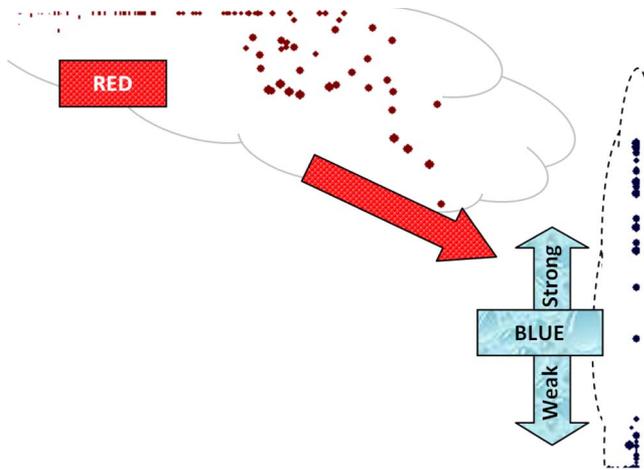


Fig. 11. Initial move after four eras of evolution:  $\rightarrow B \rightarrow R \rightarrow B$ . Blue, last evolved, adopts a defensive posture.

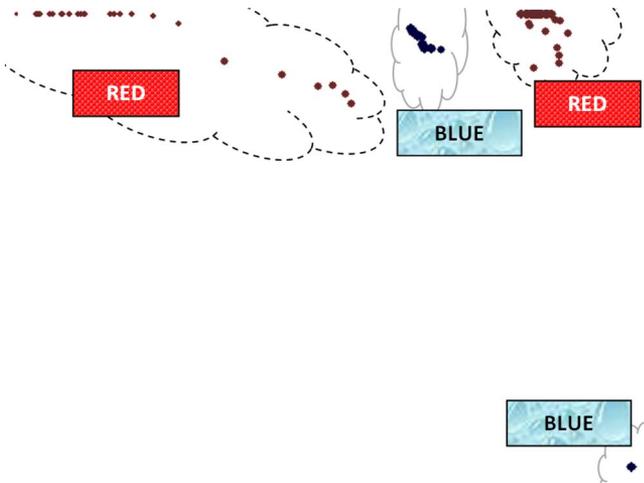


Fig. 12. With blue engaging in a defensive strategy, the red enemy goes after the nearer strong army but does not actually attack it due to its strength.

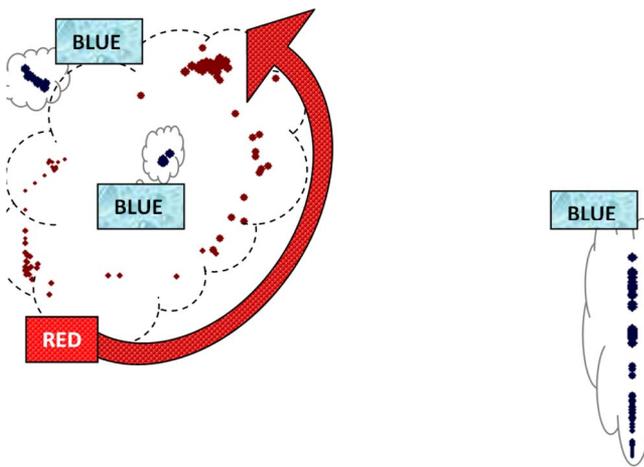


Fig. 13. Red circles blue counterclockwise.

a counterclockwise direction. The other blue cluster in the northwest corner has temporarily escaped the attention of the red agents.

d) Fig. 14. After some time has passed, all agents are now at or near maximum strength. Using a philosophy

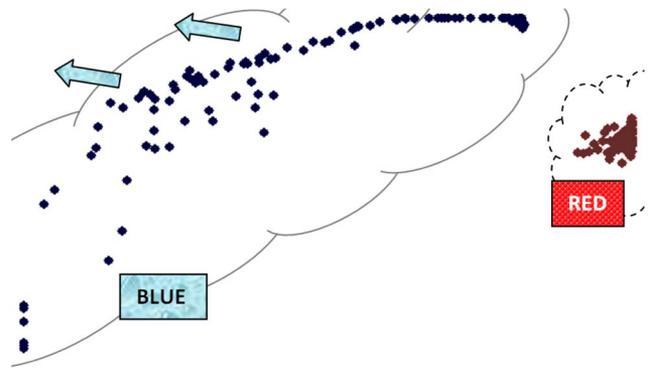


Fig. 14. Blue retreats from red.

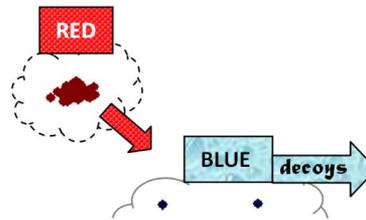
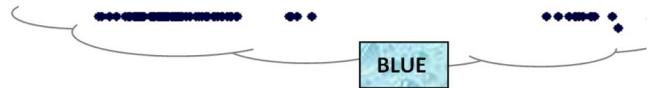


Fig. 15. Red pursues blue decoys.

that “the best offense is hiding,” the blue agents are retreating from the red agent cluster by moving to the north and west.

- e) Fig. 15. Blue decoys have broken from the group and are being pursued by the single red cluster. The other blue agents hide along the north wall. The red continues to chase the blue decoys but never captures them. In this way, blue has evolved a retreat strategy that allows it to survive for a very long time.

Fig. 16 shows the functions for the blue’s defensive logic. All values for attraction to enemy red agents are negative, so blue will always try to run from the red enemy. However, blue’s propensity to run is tempered by its strength which allows the separation shown in Fig. 11.

3) Aggressive strategy

The defensive strategy depended on the opponent blue not being willing to attack strong red agents. Additional generations of evolution of the red army address this. The evolution eras are now  $\rightarrow B \rightarrow R \rightarrow B \rightarrow R$ . When given a chance to optimize against the defensive strategy, an aggressive strategy arose. The defensive blue strategy, effective in  $\rightarrow B \rightarrow R \rightarrow B$ , remains unchanged. The evolution of the red agents, though, has made the blue strategy ineffective. Here are some snapshots of the action. Also see video #5 [72].

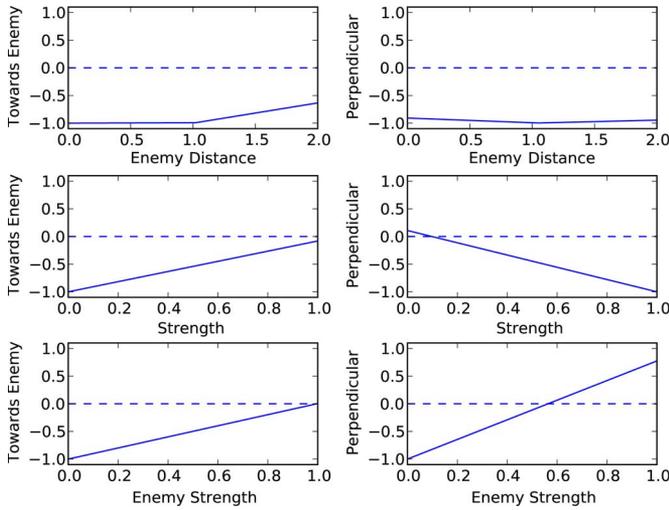


Fig. 16. Functions for the blue team engaged in a defensive strategy. Figs. 6–9 show this strategy as blue against an aggressive red strategy.

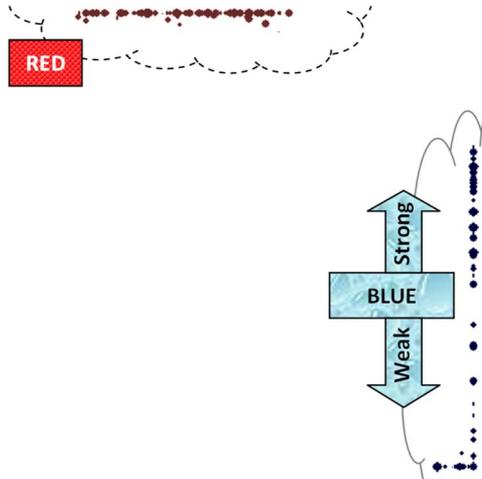


Fig. 17. Red uses an aggressive strategy against blue’s defensive strategy. While the defensive strategy divides its army, the aggressive strategy pushes against the top wall and moves toward the upper left corner. Sequential snapshots of the conflict are shown in Figs. 18 and 19.

- a) Fig. 17. The red agents move along the north wall to the right. There is some separation of the blue agents. Mostly, stronger blue agents move up the east wall, and some of the weaker blue agents move down.
- b) Fig. 18. The red agents make a sharp right turn at the northeast corner and move toward the meek blue agents who begin to retreat down the east wall.
- c) Fig. 19. The red agents move downward on the east wall. The strong agents lead the way. The red agents catch the defensive agents in the bottom left corner and rapidly kill them.

Fig. 20 shows the functions used to control the red aggressive agents. The actuator functions for the blue remain those shown in Fig. 16. Rather than avoiding strong agents, the red agents aggressively attack them to good effect. The propensity for moving toward enemy agents is related to an agent’s strength. The constant value of 1.0 for perpendicular movement based on strength results in the wall following behavior.

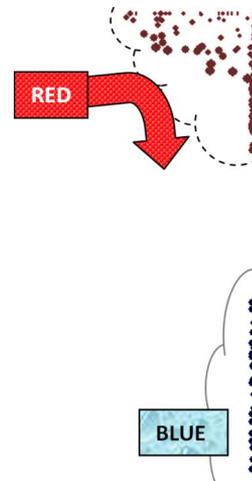


Fig. 18. Red uses an aggressive strategy against blue’s defensive strategy. The defensive blue strategy retreats to the corner, while the aggressive red strategy follows. Note how the weaker red elements are safely protected behind the army.

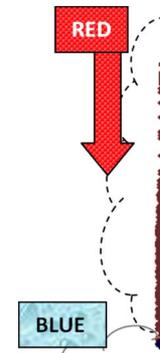


Fig. 19. Red uses an aggressive strategy against blue’s defensive strategy. Blue attempts to hide in the corner while red’s entire army comes upon it, resulting in a quick red victory.

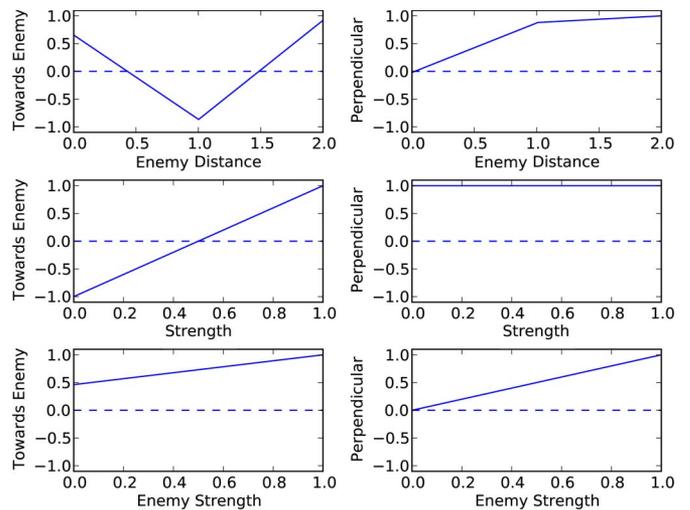


Fig. 20. Actuator functions for the red team engaged in an aggressive strategy. These actuator functions are the result of evolving the red team against the defensive blue team strategy whose actuator functions are shown in Fig. 16. The initialization is the same as shown in Fig. 6. The resulting performance is illustrated in Figs. 17–19.

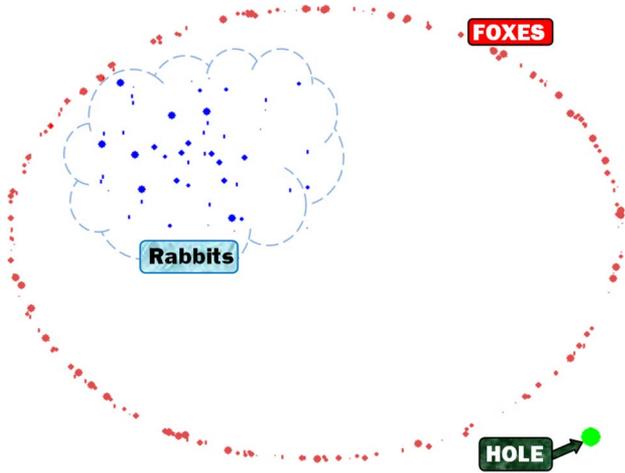


Fig. 21. Starting position for the foxes and rabbits. The foxes are red, while the rabbits are blue. The green circle is the hole.

C. Foxes Versus Rabbits

A third example used to illustrate swarm inversion using disjunctive Combs control is a contest dubbed “foxes versus rabbits.” In this model, a husk of rabbits attempts escape to a hole guarded by a skulk of predator foxes. Like that in the war model, each rabbit and fox agent is assigned a strength which determines whether or not it will survive in combat. However, unlike before, strength neither degrades nor increases over time. There are 200 foxes and 50 rabbits.

The sensors used in this model are as follows:

- 1)  $S_1$  = distance to the center of friendly agents;
- 2)  $S_2$  = distance to the nearest opponent agent;
- 3)  $S_3$  = distance to the exit (hole);
- 4)  $S_4$  = difference between self-strength and nearest opponent’s strength.

Fitness is defined as

$$C = \sum_{i=1}^T (10e_i + s_i) \tag{7}$$

where  $e_i$  is the number of rabbits who have escaped by time  $i$  and  $s_i$  is the number of rabbits who have neither escaped nor been killed and at time  $T = 2000$ .

The behaviors observed are as follows.

- 1) *Rush*. First, the foxes are evolved, and then, the rabbits are evolved. Thus, the scenario is

$$\rightarrow F \rightarrow R.$$

The performance snapshots are shown in Figs. 21–24. In this simplest case, the foxes head for the rabbits, and the rabbits run away from the foxes. The rabbits have not yet learned to run to the hole.

- a) Fig. 21. This is the initialization of the contest.
- b) Fig. 22. As the foxes close in, the rabbits move toward the hole in order to escape from the foxes.
- c) Fig. 23. The foxes continue to push the rabbits into the hole. The rabbits are not actually trying to move toward the hole but rather away from the foxes. This

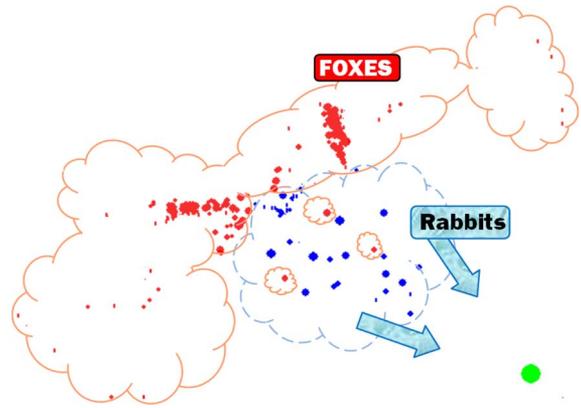


Fig. 22. Rabbits head for the hole, while the foxes chase after them.

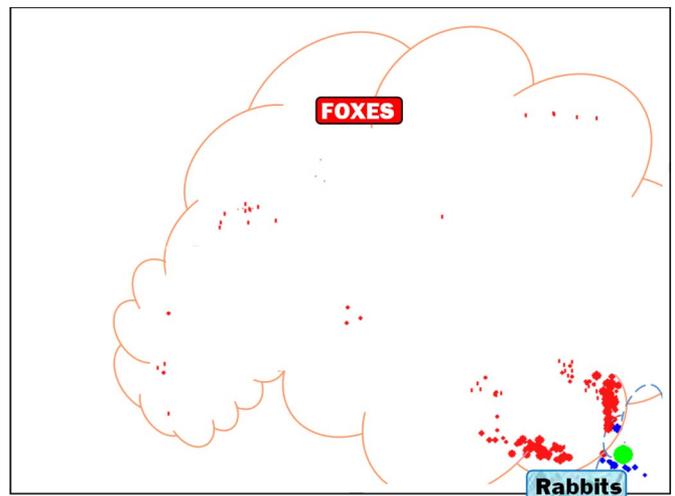


Fig. 23. Rabbits head for the hole, while the foxes chase after them.

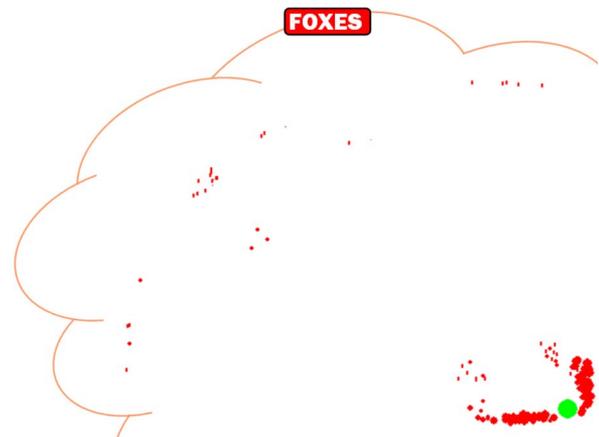


Fig. 24. All rabbits have either escaped or been caught.

happens to cause them to move close to the hole, allowing many to escape.

- d) Fig. 24. The simulation ends quickly as the foxes kill any nonescaping rabbits.
- 2) *Confusion*. The foxes are next evolved to counter the *Rush* scenario

$$\rightarrow F \rightarrow R \rightarrow F.$$

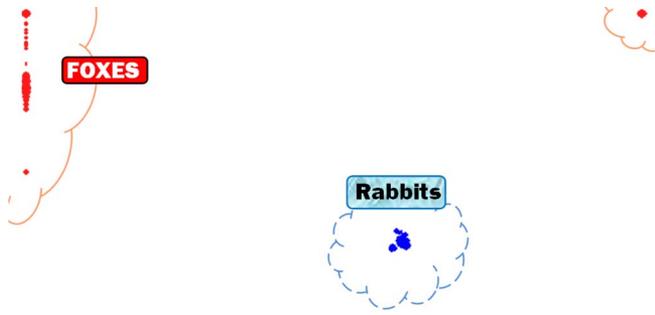


Fig. 25. Rabbits are confused and thus sit in the middle rather than running for the hole.

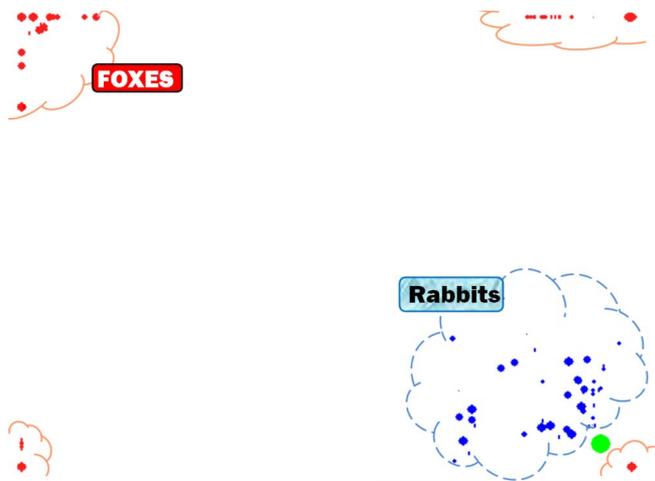


Fig. 26. Foxes hang around the outside corners, while the rabbits make it directly to the hole.

Since the rabbits were running away from the foxes rather than toward the hole, when the foxes were evolved against this behavior, they developed a strategy of confusion. The foxes occupy the corners of the area, thus keeping the rabbits in the center. This is shown in Fig. 25.

3) *Confusion resolution.* The rabbits were again evolved

$$\rightarrow F \rightarrow R \rightarrow F \rightarrow R.$$

Predictably, the rabbits head toward the target unimpeded while the foxes, still applying their confusion tactics, remain huddled in the corners. This is shown in Fig. 26. However, the agents are not actually trying to move toward the hole. Instead, they are heading to the nearest foxes who are behind the hole.

4) *Flanking.* Evolution continued. At the end of a rabbit evolution, deception and flanking emerged as a winning strategy. The repeated evolutions resulted in clever rabbit behavior. As the foxes placed themselves between the rabbits and the hole, the rabbits move away from the foxes, thereby drawing the foxes further from the hole. The rabbits then sneak little by little around the foxes and

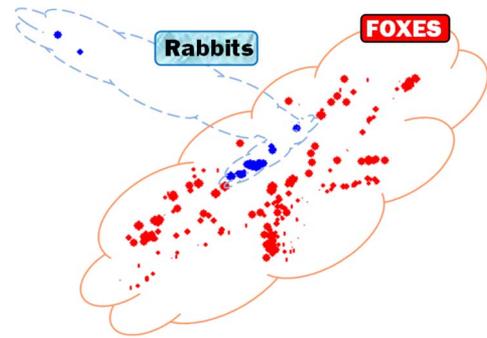


Fig. 27. In the initial stages of the flanking strategy, the rabbits wait for the foxes to approach.

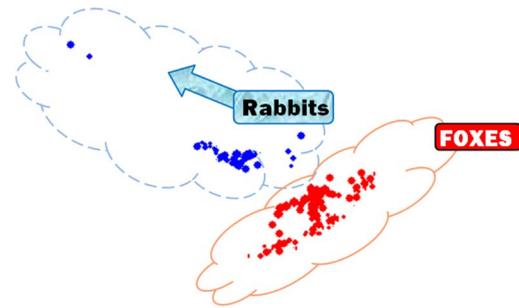


Fig. 28. After the foxes have gotten close enough in the flanking strategy, the rabbits run, leaving the foxes in a clump.

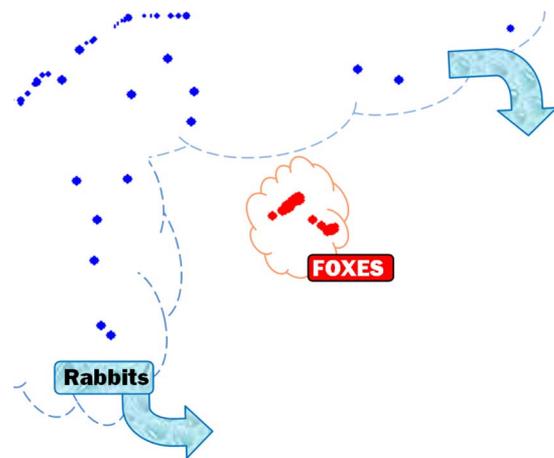


Fig. 29. Rabbits head both directions around the foxes.

make their way to the rabbit hole. Snapshots are shown in Figs. 27–31.

- a) Fig. 27. The rabbits allow themselves to be surrounded.
- b) Fig. 28. Just before the foxes close in for the kill, the rabbits break out moving away from the hole, leaving the foxes in a clump.
- c) Fig. 29. At the northwest corner, the rabbits bifurcate and begin to run around the flanks of the fox clump.

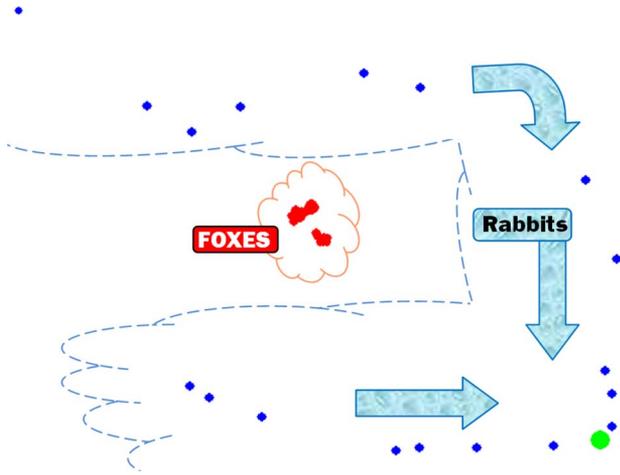


Fig. 30. Since the foxes are distracted by the nearest agents, the other rabbits sneak to the hole.

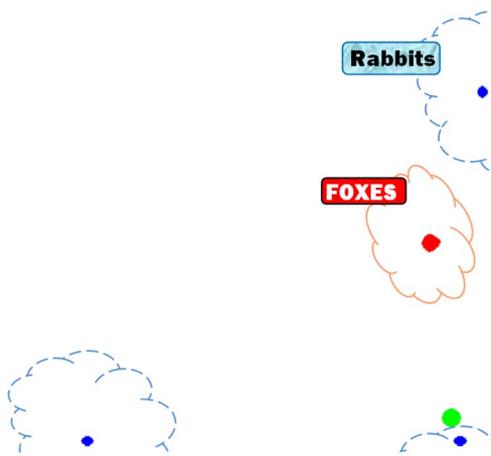


Fig. 31. Eventually, there are insufficient rabbits to attract the attention, and the foxes move close to the hole, preventing any future rabbit escapes.

- d) Fig. 30. The sneaky rabbits then manage to make it to the exit. This works because the foxes only have a sense of the nearest rabbit. The flanking rabbits are, in this sense, invisible to the foxes.
- e) Fig. 31. Eventually, the foxes detect the flanking rabbits. The strategy of the rabbits then ceases to work, and the foxes move to a better defensive position. The remaining rabbits are now blocked from the hole and are doomed to a life above ground.

V. CONCLUSION

Combs control of sensors on swarm agents adjusted in accordance to swarm performance is a methodology which can reveal interesting and unexpected emergent swarm properties. This paper, to our knowledge, is the first to explore such generalized experimental investigation of such properties. Although there is a (stochastic) fitness used in the evolution of the swarm, the aim is not fitness maximization but, rather, observation of emergent behavior as the swarm was evolved. Once the swarm was evolved, the relatively transparent explanation facility characteristic of Combs control can be examined to help understand

why the emergent behavior occurs. Combs control also has the advantage of requiring fewer tunable parameters, thereby reducing the dimensionality of the search space.

There are numerous variations of swarm inversion not yet considered.

- 1) Application of coevolution [61] can potentially evolve strategies superior to a large number of strategies rather than in response to a single opponent strategy.
- 2) Each agent can have a different control mechanism. Evolution can then be applied within a single simulation wherein killed agents are immediately replaced by replication of a mutated agent more fit.
- 3) Agents can be fitted with adaptive states as is the case, for example, when worker ants are recruited to be soldier ants when the colony is under attack [16].

Graceful degradation of performance with loss of agent count as promised by property characteristic at the swarm level has yet to be empirically tested as has graceful degradation of swarm performance due to loss of sensors at the agent level as promised by Combs control. Fitness in the swarm evolution is determined in many cases by the evolved strategies of the opponents. In order for evolved strategies to be placed into practice as viable tactical options, the robustness of the strategies must be tested on perturbations of the fitness. As witnessed by the necessity of the use of the Linux cluster, these and other studies involving swarm inversion are computationally intensive. Development of gaming models to capture the emergent properties theoretically would be useful in circumventing the need for this overhead. Although theoretical properties such as stability [39] or a swarm of distributed wireless agents converging to a Nash equilibrium [37], [83] are useful, the *tactics* by which properties are achieved, e.g., flanking, self-sacrifice, deception, and protecting the weak, are the focus of the current work.

Swarm inversion can result in effective survival strategies in swarm games. Through a process of evolutionary tuning, simple rules in agents allow emergent behavior that extends survival time for the swarms. More sophisticated simulations could find use in the development and experimental analysis of swarming in business models [17], military tactics [10], [35], finance [62], social science [90], and game theory [93].

Videos of the simulations discussed in this paper are available online and are highly instructive in the visualization of the interesting emergent behaviors discussed [72]. Documented Python code for all of the simulations is also available on the same site. Since all simulations contain stochastic components, running the code does not exactly replicate the contents of the video. Our experience, though, is that the general emergent tactics remain consistent.

APPENDIX

The emergent behaviors resulting from the three simple swarm rules in Section I are as follows.

- 1) This is a simple model used for termites stacking wood or ants collecting their dead into piles [16].
- 2) The emergent behavior is attraction of all the agents to a fixed set of points. If, in the coupling, all agents can be

linked in the rules to all other agents, convergence is to a single point.

### 3) All agents diffusively disperse.

The problems in 2) and 3) were proposed by Bonabeau and Meyer [17] and analyzed by Gravagne and Marks [39]. Additional details and videos illustrating the behaviors of all three of these swarms are available on NeoSwarm.com.

## REFERENCES

- [1] S. Abe, *Support Vector Machines for Pattern Classification*, 2nd ed. New York: Springer-Verlag, 2010.
- [2] B. Adenso-Diaz and M. Laguna, "Fine-tuning of algorithms using fractional experimental designs and local search," *Oper. Res.*, vol. 54, no. 1, pp. 99–114, Jan./Feb. 2006.
- [3] M. E. Aggoune, L. E. Atlas, D. A. Cohn, M. J. Damborg, M. A. El-Sharkawi, and R. J. Marks, II, "Artificial neural networks for power system static security assessment," in *Proc. IEEE Int. Symp. Circuits Syst.*, Portland, OR, May 8–11, 1989, pp. 490–494.
- [4] C. Ansótegui, M. Sellmann, and K. Tierney, "A gender-based genetic algorithm for the automatic configuration of solvers," in *Proc. CP*, vol. 5732, LNCS, I. P. Gent, Ed., Heidelberg, Germany, pp. 142–157, Springer.
- [5] P. Arabshahi, J. J. Choi, R. J. Marks, II, and T. P. Caudell, "Fuzzy control of backpropagation," in *Proc. 1st IEEE Int. Conf. Fuzzy Syst.*, San Diego, CA, Mar. 1992, pp. 967–972.
- [6] P. Arabshahi, R. J. Marks, II, and T. P. Caudell, "Adaptation of fuzzy inferencing: A survey," in *Proc. IEEE/Nagoya University WWW Learn. Adapt. Syst.*, Nagoya, Japan, Oct. 22/23, 1993, pp. 1–9.
- [7] P. Arabshahi, J. J. Choi, R. J. Marks, II, and T. P. Caudell, "Fuzzy parameter adaptation in optimization: Some neural net training examples," *IEEE Comput. Sci. Eng.*, vol. 3, no. 1, pp. 57–65, 1996.
- [8] P. Arabshahi, R. J. Marks, II, S. Oh, T. P. Caudell, J. J. Choi, and B. G. Song, "Pointer adaptation and pruning of min–max fuzzy estimation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 9, pp. 696–709, Sep. 1997.
- [9] P. Arabshahi, A. Gray, I. Kassabalidis, A. Das, S. Narayanan, M. El-Sharkawi, and R. J. Marks, II, "Adaptive routing in wireless communication networks using swarm intelligence," in *Proc. 19th AIAA Int. Commun. Satell. Syst. Conf.*, Toulouse, France, Apr. 17–20, 2001.
- [10] J. Arquilla and D. Ronfeldt, *Swarming and the Future of Conflict*. Santa Monica, CA: RAND Corp., 2000.
- [11] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*, 2nd ed. San Diego, CA: Academic, 2012.
- [12] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation—The New Experimentalism*. Berlin, Germany: Springer-Verlag, 2006.
- [13] R. Beckers, O. E. Holland, and J. L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Prog. Artificial Life IV*. Cambridge, MA: MIT Press, 1994, pp. 181–189.
- [14] M. Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*. Berlin, Germany: Springer-Verlag, 2009.
- [15] B. E. Bishop, "On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 5, pp. 608–615, Sep. 2003.
- [16] E. Bonabeau, M. Dorigo, and G. Thereulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.
- [17] E. Bonabeau and C. Meyer, "Swarm intelligence, a whole new way to think about business," *Harv. Bus. Rev.*, vol. 79, no. 5, pp. 106–114, May 2001.
- [18] C. Calude, *Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View*. Berlin, Germany: Springer-Verlag, 2001.
- [19] O. Castillo and W. Pedrycz, *Soft Computing for Intelligent Control and Mobile Robotics*. Berlin, Germany: Springer-Verlag, 2010.
- [20] J. J. Choi, P. Arabshahi, R. J. Marks, II, and T. P. Caudell, "Fuzzy parameter adaptation in neural systems," in *Proc. Int. Conf. Neural Netw.*, Baltimore, MD, Jun. 1992, pp. 232–238.
- [21] W. E. Combs, "Reconfiguring the fuzzy rule matrix for large time-critical applications," in *Proc. 3rd Annu. Int. Conf. Fuzzy-Neural Appl., Syst., Tools*, Nashua, NH, Nov. 1995, pp. 18:1–18:7.
- [22] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 1–11, Feb. 1998.
- [23] W. E. Combs, J. J. Weinschenk, and R. J. Marks, II, "Genomic systems design: A novel, biologically-based framework for enhancing the adaptive, autonomous capabilities of computer systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Budapest, Hungary, Jul. 25–29, 2004, pp. 1491–1496.
- [24] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, no. 1, pp. 317–365, Aug. 1998.
- [25] A. K. Das, R. J. Marks, II, M. A. El-Sharkawi, P. Arabshahi, and A. Gray, "The minimum power broadcast problem in wireless networks: An ant colony system approach," in *Proc. IEEE CAS Workshop Wireless Commun. Netw.*, Pasadena, CA, Sep. 5–6, 2002.
- [26] S. Das, U. Halder, and D. Maity, "Chaotic dynamic characteristics of social foraging swarms—An analysis," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1288–1293, Aug. 2012.
- [27] P. Dasgupta, "A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 3, pp. 549–563, May 2008.
- [28] W. A. Dembski and R. J. Marks, II, "Conservation of information in search: Measuring the cost of success," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 5, pp. 1051–1061, Sep. 2009.
- [29] W. A. Dembski and R. J. Marks, II, "Bernoulli's principle of insufficient reason and conservation of information in computer search," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Antonio, TX, Oct. 2009, pp. 2647–2652.
- [30] W. A. Dembski and R. J. Marks, II, "The search for a search: Measuring the information cost of higher level search," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 14, no. 5, pp. 475–486, 2010.
- [31] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
- [32] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
- [33] R. W. Duren, R. J. Marks, II, P. D. Reynolds, and M. L. Trumbo, "Real-time neural network inversion on the SRC-6e reconfigurable computer," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 889–901, May 2007.
- [34] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evol. Comput.*, 2000, vol. 1, pp. 84–88.
- [35] S. J. A. Edwards, *Swarming on the Battlefield Past, Present, and Future*. Santa Monica, CA: RAND Corp., 2000.
- [36] J. E. Ervin and S. E. Altekin, "Combs method used in intuitionistic fuzzy logic application," in *Proc. Appl. Fuzzy Sets Theory: 7th Int. Workshop Fuzzy Logic*, F. Masulli, S. Mitra, and G. Pasi, Eds., 2007, pp. 306–312, Springer.
- [37] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 463–476, May 2006.
- [38] A. S. Fukunaga, "Automated discovery of local search heuristics for satisfiability testing," *Evol. Comput.*, vol. 16, no. 1, pp. 31–61, 2008.
- [39] I. A. Gravagne and R. J. Marks, II, "Emergent behaviors of protector, refugee and aggressor swarm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 471–476, Apr. 2007.
- [40] M. Günes, U. Sorges, and I. Bouazizi, "ARA—The ant-colony based routing algorithm for MANETs," in *Proc. IEEE ICPPW*, 2002, pp. 79–85.
- [41] M. Hadzic and E. Chang, "Application of digital ecosystem design methodology within the health domain," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 4, pp. 779–788, Jul. 2010.
- [42] L. H. Hamel, *Knowledge Discovery With Support Vector Machines*. Hoboken, NJ: Wiley, 2009.
- [43] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization," in *Proc. IEEE/RSJ Int. Conf. IROS*, Oct. 2001, pp. 1073–1087.
- [44] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy, "An experimental investigation of model-based parameter optimisation: SPO and beyond," in *Proc. GECCO*, F. Rothlauf, Ed., New York, pp. 271–278, ACM press.
- [45] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, "ParamILS: An automatic algorithm configuration framework," *J. Artif. Intell. Res.*, vol. 36, pp. 267–306, 2009.
- [46] J. N. Hwang, C. H. Chan, and R. J. Marks, II, "Frequency selective surface design based on iterative inversion of neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, San Diego, CA, Jun. 17–21, 1990, vol. 1, pp. 39–44.
- [47] J. N. Hwang, J. J. Choi, S. Oh, and R. J. Marks, II, "Query based learning applied to partially trained multilayer perceptrons," *IEEE Trans. Neural Netw.*, vol. 2, no. 1, pp. 131–136, Jan. 1991.

- [48] A. Ilichinski, *Artificial War: Multiagent-Based Simulation of Combat*. Singapore: World Scientific, 2004.
- [49] M. Jamshidi, N. Vadiee, and T. Ross, *Fuzzy Logic and Control: Software and Hardware Applications*, vol. 2. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [50] B. Jayaram, "On the law of importation  $(x \wedge y) \rightarrow z \equiv (x \rightarrow (y \rightarrow z))$  in fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 130–144, Feb. 2008.
- [51] C. A. Jensen, R. D. Reed, M. A. El-Sharkawi, and R. J. Marks, II, "Location of operating points on the dynamic security border using constrained neural network inversion," in *Proc. ISAP*, Seoul, Korea, Jul. 6–10, 1997, pp. 209–217.
- [52] C. A. Jensen, R. D. Reed, R. J. Marks, II, M. A. El-Sharkawi, J.-B. Jung, R. T. Miyamoto, G. M. Anderson, and C. J. Eggen, "Inversion of feedforward neural networks: Algorithms and applications," *Proc. IEEE*, vol. 87, no. 9, pp. 1536–1549, Sep. 1999.
- [53] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks, P. Asabshahi, and A. A. Gray, "Swarm intelligence for routing in communication networks," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001, pp. 3613–3617.
- [54] I. Kassabalidis, M. El-Sharkawi, R. J. Marks, II, P. Arabshahi, and A. Gray, "Adaptive-SDR: Adaptive swarm-based distributed routing," in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, May 12–17, 2002, pp. 351–354.
- [55] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. IV, pp. 1942–1948.
- [56] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1997, pp. 303–308.
- [57] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.
- [58] H. Leung, R. Kothari, and A. Minai, "Phase transition in a swarm algorithm for self-organizing construction," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 4, pp. 046111-1–046111-9, Oct. 2003.
- [59] S. Li, D. C. Wunsch, E. O'Hair, and M. G. Giesselmann, "Extended Kalman filter training of neural networks on a SIMD parallel machine," *J. Parallel Distrib. Comput.*, vol. 62, no. 4, pp. 544–562, Apr. 2002.
- [60] F. Lin and Y.-H. Pai, "Using multi-agent simulation and learning to design new business processes," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 3, pp. 380–384, May 2000.
- [61] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Proc. Congr. Evol. Comput.*, 2001, vol. 2, pp. 1101–1108.
- [62] F. Luna and A. Perrone, *Agent-Based Methods in Economics and Finance: Simulations in Swarm*. Norwell, MA: Kluwer, 2002.
- [63] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, no. 1, pp. 1–13, Jan. 1975.
- [64] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *Int. J. Man-Mach. Stud.*, vol. 8, no. 6, pp. 669–678, Nov. 1976.
- [65] R. J. Marks, II, S. Oh, P. Arabshahi, T. P. Caudell, J. J. Choi, and B. G. Song, "Steepest descent of min-max fuzzy if-then rules," in *Proc. Int. Joint Conf. Neural Netw.*, Beijing, China, Nov. 3–6, 1992, vol. III, pp. 471–477.
- [66] R. J. Marks II, Ed., *Fuzzy Logic Technology and Applications*. Piscataway, NJ: IEEE Press, 1994.
- [67] A. Montresor, H. Meling, and Ö. Babaoglu, "Messor: Load-balancing through a swarm of autonomous agents," Dept. Comput. Sci., Univ. Bologna, Bologna, Italy, Tech. Rep. UBLCS-2002-11, 2002.
- [68] L. S. Moulin, A. P. A. da Silva, M. A. El-Sharkawi, and R. J. Marks, II, "Neural network and support vector machines applied to power systems transient stability analysis," *Int. J. Eng. Intell. Syst. Elect. Eng. Commun.*, vol. 9, no. 4, pp. 205–212, Dec. 2001.
- [69] L. S. Moulin, A. P. Alves da Silva, M. A. El-Sharkawi, and R. J. Marks, II, "Support vector and multilayer perceptron neural networks applied to power systems transient stability analysis with input dimensionality reduction," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, 2002, pp. 1308–1313.
- [70] L. S. Moulin, A. P. A. da Silva, M. A. El-Sharkawi, and R. J. Marks, II, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 818–825, May 2004.
- [71] V. Nannen and A. E. Eiben, "Relevance estimation and value calibration of evolutionary algorithm parameters," in *Proc. IJCAI*, Menlo Park, CA, 2007, pp. 975–980.
- [72] Neoswarm War Games. [Online]. Available: [http://neoswarm.com/war\\_games.html](http://neoswarm.com/war_games.html)
- [73] S. Oh, R. J. Marks, II, and M. A. El-Sharkawi, "Query based learning in a multilayered perceptron in the presence of data jitter," in *Proc. 1st Int. Forum Appl. Neural Netw. Power Syst.*, Seattle, WA, Jul. 23–26, 1991, pp. 72–75.
- [74] M. Oltean, "Evolving evolutionary algorithms using linear genetic programming," *Evol. Comput.*, vol. 13, no. 3, pp. 387–410, Sep. 2005.
- [75] D. Prokhorov, R. Santiago, and D. C. Wunsch, "Adaptive critic designs: A case study for neurocontrol," *Neural Netw.*, vol. 8, no. 9, pp. 1367–1372, Dec. 1995.
- [76] R. D. Reed, S. Oh, and R. J. Marks, II, "Regularization using jittered training data," in *Proc. Int. Joint Conf. Neural Netw.*, Baltimore, MD, Jun. 1992, pp. 147–152.
- [77] R. D. Reed, R. J. Marks, II, and S. Oh, "An equivalence between sigmoidal gain scaling and training with noisy (jittered) input data," in *Proc. RNNS/IEEE Symp. Neuroinformat. Neurocomput.*, Rostov-on-Don, Russia, Oct. 1992, pp. 120–127.
- [78] R. D. Reed, R. J. Marks, II, and S. Oh, "Similarities of error regularization, sigmoid gain scaling, target smoothing and training with jitter," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 529–538, May 1995.
- [79] R. D. Reed and R. J. Marks, II, *Neural Smoothing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press, 1999.
- [80] P. D. Reynolds, R. W. Duren, M. L. Trumbo, and R. J. Marks, II, "FPGA implementation of particle swarm optimization for inversion of large neural networks," in *Proc. IEEE SIS*, Pasadena, CA, Jun. 8–10, 2005, pp. 389–392.
- [81] E. W. Saad and D. C. Wunsch, II, "Neural network explanation using inversion," *Neural Netw.*, vol. 20, no. 1, pp. 78–93, Jan. 2007.
- [82] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2001.
- [83] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, Feb. 2002.
- [84] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1998, pp. 69–73.
- [85] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proc. 7th EP*, 1998, pp. 591–600.
- [86] A. Shklarsh, G. Ariel, E. Schneidman, and E. Ben-Jacob, "Smart swarms of bacteria-inspired agents with performance adaptable interactions," *PLoS Comput. Biol.*, vol. 7, no. 9, p. e1002177, Sep. 2011.
- [87] W. Siler and J. J. Buckley, *Fuzzy Expert Systems and Fuzzy Reasoning*. New York: Wiley, 2004.
- [88] B. G. Song, R. J. Marks, II, S. Oh, P. Arabshahi, T. P. Caudell, and J. J. Choi, "Adaptive membership function fusion and annihilation in fuzzy if-then rules," in *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, Mar. 1993, vol. II, pp. 961–967.
- [89] R. J. Streifel, R. J. Marks, II, R. Reed, J. J. Choi, and M. Healy, "Dynamic fuzzy control of genetic algorithm parameter coding," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 3, pp. 426–433, Jun. 1999.
- [90] P. Terna, "Simulation tools for social scientists: Building agent based models with SWARM," *J. Artif. Soc. Social Simul.*, vol. 1, no. 2, pp. 1–12, 1998.
- [91] N. Todorovic and S. Petrovic, "Bee colony optimization algorithm for nurse rostering," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 43, no. 2, pp. 467–473, Mar. 2013.
- [92] B. B. Thompson, R. J. Marks, II, M. A. El-Sharkawi, W. J. Fox, and R. T. Miyamoto, "Inversion of neural network underwater acoustic model for estimation of bottom parameters using modified particle swarm optimizers," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, Jul. 20–24, 2003, pp. 1301–1306.
- [93] K. Tumer and D. H. Wolpert, *Collectives and the Design of Complex Systems*. New York: Springer-Verlag, 2004.
- [94] J. J. Weinschenk, W. E. Combs, and R. J. Marks, II, "Avoidance of rule explosion by mapping fuzzy systems to a disjunctive rule configuration," in *Proc. Int. Conf. Fuzzy Syst.*, St. Louis, MO, May 25–28, 2003, pp. 43–48.
- [95] J. J. Weinschenk, R. J. Marks, II, and W. E. Combs, "Layered URC fuzzy systems: A novel link between fuzzy systems and neural network," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, Jul. 20–24, 2003, pp. 2995–3000.
- [96] J. J. Weinschenk, R. J. Marks, II, and W. E. Combs, "On the use of Fourier methods in URC fuzzy system design," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Budapest, Hungary, Jul. 25–29, 2004, vol. 2, pp. 911–916.

- [97] J. J. Weinschenk, W. E. Combs, and R. J. Marks, II, "On the avoidance of rule explosion in fuzzy inference engines," *Int. J. Inf. Technol. Intell. Comput.*, vol. 1, no. 4, pp. 1–26, 2007.
- [98] Z. Yuan, M. de Oca, M. Birattari, and T. Stützle, "Modern continuous optimization algorithms for tuning real and integer algorithm parameters," in *Swarm Intelligence*, M. Dorigo, M. Birattari, G. Di Caro, R. Doursat, A. Engelbrecht, D. Floreano, L. Gambardella, R. Gross, E. Sahin, H. Sayama, and T. Stützle, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 203–214.



**Winston Ewert** received the B.Sc. degree in computer science from Trinity Western University, Langley, BC, Canada, in 2009 and the M.S. degree in computer science from Baylor University, Waco, TX, in 2010, where he is currently working toward the Ph.D. degree.

His research interests include search algorithms, optimization, and emergent behavior.



**Robert J. Marks, II** (S'71–M'72–SM'83–F'94) is a Distinguished Professor of electrical and computer engineering with Baylor University, Waco, TX. His consulting activities include Microsoft Corporation, Pacific Gas and Electric Company, and Boeing Computer Services. His research has been funded by organizations such as the National Science Foundation, General Electric, Southern California Edison, the Electric Power Research Institute (EPRI) the Air Force Office of Scientific Research, the Office of Naval Research, the Whitaker Foundation, Boeing

Defense, Space and Security, the National Institutes of Health, the Jet Propulsion Laboratory, the Army Research Office, and the National Aeronautics and Space Administration (NASA).

Dr. Marks II is a fellow of The Optical Society of America. He was the recipient of a NASA Tech Brief Award and a Best Paper Award from the American Brachytherapy Society for prostate cancer research. He was also the recipient of the IEEE Outstanding Branch Counselor Award, the IEEE Centennial Medal, the IEEE Neural Networks Society Meritorious Service Award, the IEEE Circuits and Systems Society Golden Jubilee Award, and the 2007 Volunteer of the Year Award from the IEEE Computational Intelligence Society Chapter of the IEEE Dallas Section. He was named a Distinguished Young Alumnus of Rose-Hulman Institute of Technology, is an inductee into the Texas Tech Electrical Engineering Academy, and was awarded the Banned Item of the Year in 2007 from the Discovery Institute. He served for 17 years as the Faculty Advisor to the University of Washington Chapter of Campus Crusade for Christ.



**Benjamin B. Thompson** received the B.S. degree from the School of Engineering and Computer Science, Baylor University, Waco, TX, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 2002 and 2004, respectively.

He is currently a Research Engineer with the Undersea Weapons Office, Applied Research Laboratory, The Pennsylvania State University, State College, where he is also the Head of the Tactical Processing Department. His research interests include swarm intelligence, autonomous undersea vehicles, sonar signal processing, multiagent systems, intelligent control systems, and machine learning.



**Albert Yu** received the B.S. degree in bioengineering from Rice University, Houston, TX, in 2007 and the M.S. degree in electrical and computer engineering from Baylor University, Waco, TX, in 2011. He is currently working toward the Ph.D. degree at the University of Washington, Seattle.

His research interests include multiagent system modeling and multiobjective optimization.