Proceedings of IMEC 2005:
ASME INTERNATIONAL MECHANICAL ENGINEERING CONGRESS AND EXPOSITION
November 5-11, 2005 Orlando, Florida

IMEC2005-81890

THE EFFECTIVENESS OF TRANSFERRING MULTI-AGENT BEHAVIORS FROM A LEARNING ENVIRONMENT IN THE PRESENCE OF SYNTHETIC SOCIAL FEATURES

Keith Garfield
Institute for Simulation and Training, UCF

Annie Wu
School of Computer Science, UCF

Mehmet Onal
Industrial Engineering and Management Systems, UCF

Britt Crawford
School of Computer Science, UCF

Adam Campbell
School of Computer Science, UCF

Randall Shumaker
Institute for Simulation and Training, UCF

ABSTRACT

The diverse behavior representation schemes and learning paradigms being investigated within the robotics community share the common feature that successful deployment of agents requires that behaviors developed in a learning environment are successfully applied to a range of unfamiliar and potentially more complex operational environments. The intent of our research is to develop insight into the factors facilitating successful transfer of behaviors to the operational environments. We present experimental results investigating the effects of several factors for a simulated swarm of autonomous vehicles. Our primary focus is on the impact of Synthetic Social Structures, which are guidelines directing the interactions between agents, much like social behaviors direct interactions between group members in the human and animal world. The social structure implemented is a dominance hierarchy, which has been shown previously to facilitate negotiation between agents. The goal of this investigation is to investigate mechanisms adding robustness to agent behavior.

INTRODUCTION

Various approaches to producing intelligent and cooperative behavior among agents have demonstrated progress in narrowly restricted domains. No single approach has yet to emerge integrating ease of development coupled with robust and adaptable behaviors allowing agents to succeed in new environments. This is important as a unifying trait among the majority of systems aiming to produce intelligent behaviors is that the learning occurs in a training environment which is expected to produce behaviors applicable to an operational environment. We report results of experiments designed to identify factors that assist in preserving successful behaviors in new environments. These experiments are part of a larger research program into the role of Synthetic Social Structures in the development and use of multi-agent (swarm) behaviors, and as such represent intermediary results.

Our work extends work performed at NRL and later SAIC, that explored the evolution of rule sets to control the behavior of individual Autonomous Vehicles (AV) operating in a team. In the previous work, Genetic Algorithm (GA) techniques were used to develop simple reactive behaviors for the AVs, though this was not required for this study. The paradigm of synthetic social structures (SSS) was coupled with the GA techniques to investigate the effect of the solution computation in terms of time to achieve an effective solution, and fitness of the solution.

The agents in this work are non-point simulated Autonomous Vehicles (AV) operating in a flat-world environment. The AV’s are reactive agents, having behaviors generated in step-wise fashion in response to the immediate environment. Each AV behavior generator allows for resolution of multiple goals, such as avoiding collision with other AVs while searching and surveilling targets in the environment. In addition to the reactive behavior rules, a synthetic social structure is imposed upon the AV swarm. The synthetic social rules constitute a static dominance hierarchy, which affect AV behavior by requiring that AV's separate from more dominant neighboring AV's. This acts as a scattering mechanism in open terrain, and resolves positioning disputes in restricted terrain.

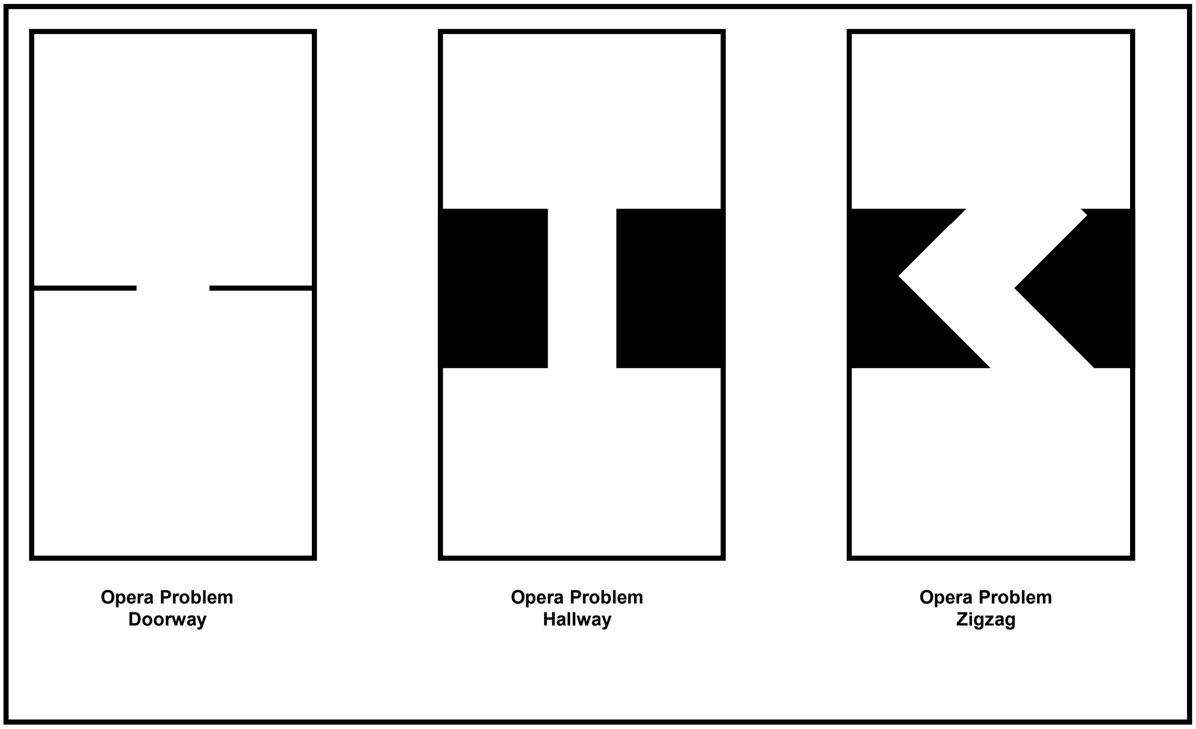

Figure 1: Opera Problem Scenarios

Three series of experiments were conducted in the simulated 2-dimensional environment. The experiments consisted of variations on the Opera Problem, which require that members of a group cooperate to negotiate a restricted passageway while moving between rooms. The Opera Problem can easily be shown to extend to the more general problem of swarm navigation in terrain with obstacles. The focus of the experiments was not pure performance, nor learning to achieve some level of capability, but maintaining capability observed in the simplest environment across more complex environments.

NOMENCLATURE

AV	Autonomous Vehicle
GA	Genetic Algorithm
PDF	Probability Distribution Field
SSS	Synthetic Social Structure

BACKGROUND

This section provides brief background information on multi-agent systems, learning and adaptation, and synthetic social structures.

Multi-Agent Systems

Multi-agent systems are those that employ multiple software/hardware agents working communally on a single mission or problem. Agents can be characterized by degree of autonomy, individual behavioral preferences, degree of global information available to them, degree of inter-agent communication [1]. In addition, agents may be classified as

cognitive or reactive. Cognitive agents deliberately develop solutions through reasoning about world state, whiles reactive agents produce emergent solutions at the system level with no individual agent demonstrating high level cognizance or capability. [1]. Multi-agent systems are being researched due to attractive properties of distributed control, (relatively) robust behaviors , and scaling of behaviors with respect to agent population [1, 2, 3].

Learning Paradigms and Adaptation

Numerous learning paradigms have been proposed in the study of intelligent agents. Among the most popular pertaining to the fields of robotics and multi-agent systems are neural networks and genetic algorithms.

The Neural Network (NN) paradigm coarsely mimics the biology of the brain, with one or more layers of formal neurons connected by synapse [3]. In a multi-layer system, one layer is typically the receiver of inputs, traditionally termed the perceptor layer, and one layer outputs the result, often labeled the execution layer [1]. Each synapse sends a weighted signal as output to the next layer or output (execution) discriminator. Neural networks learn through a series of training cases during which synaptic weights and firing thresholds are adjusted until some threshold of acceptable performance is attained

The Genetic Algorithm (GA) learning paradigm is a biologically inspired system in which potential solutions are formatted as genetic material, and acceptable solutions are evolved through a selection process involving multiple generations. A typical GA begins with a randomly generated population, each representing a potential solution to the problem being studied. These potential solutions are evaluated and those judged most fit form the basis of a new population of solutions, which is generated using genetically inspired operators.

The GA paradigm is currently enjoying application in many diverse problem domains. Key features common to the GA paradigm are a population of individuals representing potential solutions to a problem. The potential solutions are represented as genetic material, typically a binary string. A user defined fitness function evaluates the degree to which each individual succeeds in the problem domain. A selection function then selects individuals for reproduction to produce the next generation. Idealized genetic operators, such as crossover and mutation, combine and modify the parent genetic material to produce offspring representing new candidate solutions. The process is considered to be computationally expensive, and solutions are typically harvested either when a specified degree of proficiency is achieved or the rate of increase of solution effectiveness falls below a certain threshold.

Adaptation

One commonality among learning systems is the need to train the agent in a learning environment and transfer the learned behaviors to an operational environment. Neural nets, which have no intrinsic introspective capability, are noted for being "brittle" in this sense. For example, Xu, et. al. anticipated this effect for a neural network navigational system and incorporated six separate neural nets, each dedicated to a separate range of sensed parameters, in an attempt to broaden the range of the overall expertise of the navigational system [4].

Other systems acknowledge the difficulty of applying learning to an operational environment simply by allowing the agent to continue learning in the operational environment. While this approach is suitable for some applications (i.e. Via Voice's capability to tune itself to a single user over time), the prospect of unstructured learning is daunting. In addition, negative learning may occur and is not acceptable in safety-critical systems.

Synthetic Social Structures

Social rules play an integral role in determining the behavior of individual biological organisms. When two or more individuals attempt to operate successfully within the same physical space, they must take into account not just the inanimate environment, but the actions of the other individual(s). When two or more biological individuals repeatedly operate in the same space, social rules develop naturally. There are countless examples of biological social rules, though the computer science community has traditionally focused on only a few. Insect communities (ants and bees) have provided models for agents demonstrating team behavior through the interaction of relatively simple individual rules [2,3]. The emergent aggregate behavior is more complex than the simple rules directing individual actions, and does not require explicit control structures. More complex social animals have provided models for dynamically self-organizing teams of agents that exhibit more complex individual reasoning and behaviors [5]. An attractive social structure for this study is the dominance hierarchy, which provides an explicit mechanism to dynamically self-organize teams [5].

Social rules are though to provide benefits in a variety of ways [2,3,5]:

Resource Management: Social rules may be useful in resolving conflicts in which two agents compete over limited resources. In the case of dominance hierarchies, the more dominant agent receives priority in acquiring the resource.

Explanation of Behavior: Social rules may be helpful in allowing humans to understand or predict agent behavior, both intuitively and formally. Humans intuitively understand social structures found in nature, and when observing an agent are known to ascribe anthropomorphic qualities to the agent even if the agent is not intentionally mimicking biological behaviors. These two human qualities, an intuitive understanding and a tendency to anthropomorphize, imply that agent behavior purposefully mimicking biological social behaviors and structures will be more readily understood by humans operating the agents. More formally, intentionally designed behaviors embedded in a SSS may assist human designers of agents to predict productive behavior, and explicitly prohibit behavior injurious to the agent, human, or mission.

Team Roles: Social rules may be useful as a way of determining roles of agents within a multi-agent system, and for allowing dynamic re-assignment of roles within an ongoing mission.

Team Cognition: Team cognition represents a level of knowledge regarding the state of the team that is not entirely captured at the individual level. Social rules may be useful for achieving team cognition as each team members can compare the team state observable to them against known 'customs' or standards.

Efficiency: Social structures provide a major benefit in terms of efficiency in biological systems, and are expected to do the same for synthetic agents. Many systems resolve agent-agent conflicts inefficiently. Typically, negotiations between agents are performed at the time a conflict arises via explicit interactions. Negotiations are repeated between the same agents when similar situations are repeated, even though the outcome is not likely to differ from previous interactions. The existence of synthetic social structures allows for agents to self-organize with a minimal amount of explicit interaction. Agents can maintain a history of social interactions with other individuals so that future un-necessary negotiations with the same individuals are not repeated in future encounters.

Failure Tolerance: Social rules represent a layer of behavioral control on top of rules possibly developed using evolutionary techniques. This multi-layer nature of behavior representation may allow for graceful failure in the event of system degradation. Higher level social behaviors may become inoperative due to system failure, but the lower level evolved individual behaviors may prevent complete loss of capability.

Learning and Transfer of Behavior

The behaviors were designed, as opposed to learned. This is acceptable as the purpose of the experiments is to investigate factors that assist the transition of learned or designed behaviors from the learning/testing environment to the operational environment.

EXPERIMENT SETUP

The experiments were performed as a series of simulations using the MASON simulation systems developed at George Mason University [6]. Each simulation implemented a variation of the Opera Problem, in which a group of agents must cooperate to negotiate a passageway separating two rooms. Three basic scenarios were investigated, as depicted in Figure 1. The Doorway scenario represents the baseline scenario, and depicts two room separated by a simple doorway. The Hallway scenario extends the doorway into a long hallway. The Zigzag scenario requires that the AV's negotiate a bend in the hallway. These scenarios were intended to be incrementally more difficult for the AVs to negotiate in minor increments, the concept being that large environmental changes that would not allow any behaviors to transfer would yield trivial results. Three sets of experiments were performed, one investigating several parameters and one investigating the effect of social features more fully.

The remainder of this section describes the simulated AV capabilities and behavior engine, the implemented Dominance Hierarchy synthetic social structure, experimental metrics and parameters explored, and details of the MASON simulation system.

AV Capabilities and Behavior Engine

The AV's are represented in the simulation as independently operating reactive agents. Since individual and inter-agent behavior is the focus of the study, the detailed mechanical and sensing operation of the AVs were not explicitly represented. AVs have sensing capabilities to detect obstacles, other AV's and potentially other types of vehicles in the scenario. The sensors operate in a 360 degree field of view and a limited range. The AVs represent ground vehicles, and therefore can stop and are presumed to survive collisions with other AV's and obstacles (Note: the first series of experiments presumed the AV destroyed upon collision, slightly modifying success criteria).

Each AV attempts to satisfy a set of prioritized goals such that, while individual actions have a degree of uncertainty to them, a sequence of actions will tend to satisfy a large number of highest priority goals. This is achieved through the use of multiple Probability Distribution Fields (PDF). The use of PDF's to direct AV movement allows multiple goals to be addressed at each time step without reliance upon the complex logic structures often used to guide mission level behaviors. A PDF is basically an oval sliced into many pie-slice regions, with the probability that the AV will move into a specific region represented graphically by the radius of the region centerline.

The role of the PDF in goal resolution works as follows. Each PDF is associated with a single, distinct goal, such as moving to a point or avoiding collision. The default PDF is circular, with the AV has an equally likely chance of moving in any direction (we call this wandering behavior). Each goal corresponds to a distinct PDF shape with elongations representing higher probabilities, and contractions representing lower probabilities. For example, if required to move to a certain point in the field of play, the PDF will be elongated towards the point, making it more likely that the AV will move in that direction at each time step. PDF regions may also be completely restricted; such as when imminent collision must be avoided at all costs.

The individual PDF's are overlaid and merged to form a single set of probabilities for movement in a given time step. The AV is generally capable of moving into any region at any time step, but is more likely to move toward those that will satisfy AV goals. The result of many time step incremental movements will tend towards achieving AV goals. This system accounts for potentially conflicting AV goals in AV movement without complex cognitive reasoning functions. A weighting system is employed when combining multiple PDFs, effectively prioritizing the goal associated with each PDF. Weights may change as a function of time or other measurable parameter, to allow a variation in goal priorities over time. Thus, a PDF shaped to avoid an obstacle should have little weight while the obstacle is far away, but should gain prominence in the averaging process as the AV approaches the object.

Environment

The Opera Problem scenarios were investigated as a step in the overall development of the simulation and AV capabilities and characteristics. Opera Problem scenarios require that agents act cooperatively to pass through a doorway as if leaving a crowded theater (opera house). We have performed several sequences of Opera problem scenarios as part of the development of fuller AV capabilities. Three basic room configurations (terrains) were investigated, as shown in Fig. 1. The original Opera Problem involves two rooms connected by a single doorway, while the Opera Hall Problem has two rooms connected by a straight long hallway, and the Opera Zigzag Problem has two rooms connected by a zigzag hallway. The incorporation of small, incremental changes to the environment was designed to allow some opportunity for behaviors to transfer. The use of radically different environments was discarded as not likely to identify factors assisting in transferring results.

Dominance hierarchies were the Synthetic Social Structure of choice for these scenarios. This SSS has been shown in previous work to act as a scattering mechanism, and is believed to be efficient in resource allocation (ref Tomlinson). The resource in this case being access to the opening separating the two rooms. Typical multi-agent resource allocation systems are market or auction based, such as that presented by Goldberg [6]. Dominance hierarchies do not require duplicate negotiations when the outcome can be predetermined from past history [5].

Experiment 1 & 2 Parameters

- Population (Exp 1 only)
- Sensor Range
- Opening Width
- Social Structure

Experiment 3 Cases

- No Social Structure
- All Dominance Value Unique
- One Dominant AV
- Two dominant AVs

Figure 2: Experiment Parameters

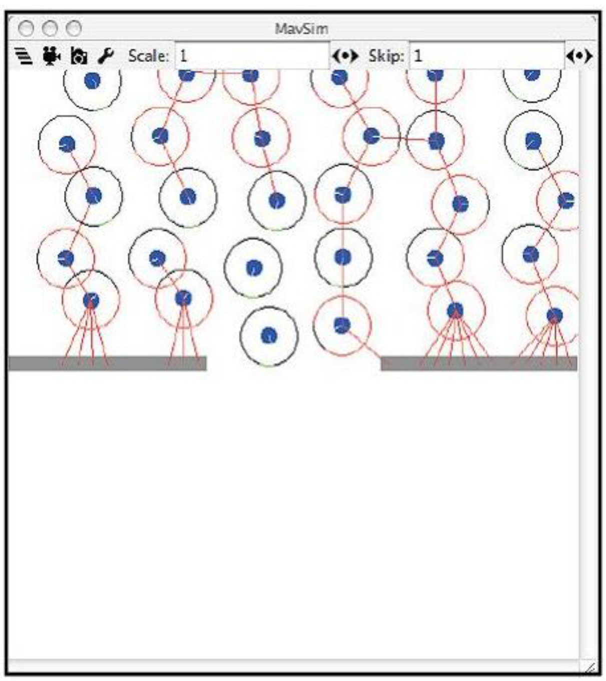

Figure 3: The MASON Simulation

The simulation environment consisted of a 2-dimensional field, separated into two areas by a barrier and connected through an opening. The configuration of the opening is dependent on the version of the problem being studied. The doorway is the simplest environment, and acts as the learning environment, or basis for the remainder of the experiment. AVs are not able to traverse off the playing field, but the edge of the field is not treated as an obstacle in the same sense as the barrier between rooms. AVs encountering the edge of the playing field simply bounce back upon collision.

Metrics/conditions/parameters

Three series of experiments were conducted. The first two series of experiments varied parameters with the goal of identifying those that most affected the capability to maintain swarm performance across the various terrains. In the first experiment, selected parameters were population size, opening width, sensor range, and the presence of social rules were investigated. As a result of the first experiment it became clear that population size was not a factor in any of our scenarios. In the second, selected parameters investigated were opening width, sensor range, and the presence of social rules. Because of the reduction in parameters reducing the number of total runs, each parameter combination was conducted 5 times and averaged.

The third series of experiments more fully investigated the effect of dominance hierarchies on the ability of the AV swarm to navigate the Hall and Zigzag configuration. The first two series indicated that the dominance SSS was a factor in improving swarm performance, but the effect was not consistent across environments. This sequence was performed to determine how the dominance SSS affected performance, and to determine if any variations of the SSS consistently improved performance across terrains. The specific cases for each series of experiments are summarized in Figure 2.

Mason

MASON is a discrete-event simulation system written in Java and deliberately designed to be compatible with the study of multi-agent (or swarm) behaviors. The system is open source, free, and available from George Mason University's Center for Social Complexity [7]. The choice of MASON for these experiments was due to program history and the ability to modify the source code to implement AV behaviors and simulation field terrain efficiently. A screen shot of the AV experiments implemented within MASON is shown in Fig. 3

RESULTS

Experimental results are summarized in this section. We stress that the purpose of the experiments was not to elicit the highest performance from the swarm, but to attempt to identify factors that allow successful behaviors and strategy in one environment to be successfully transferred to another environment. As such we are interested in consistency across environments more than success in any single environment.

Experimental results for the first series of experiments are summarized in Fig. 4. Due to a change in AV behavior and success measurement between experiments 1 and 2, we show only the results pertaining to population to eliminate confusing comparisons between experiment 1 and 2 data for other parameters. In this experiment, AV collisions resulted in destruction of the AV, so the results show surviving vehicles reaching the target as a percentage of overall population. The most consistent result of the first series of experiments was that a change in population did not significantly affect overall swarm behavior. This is a reasonable result, as one of the perceived strengths of multi-agent models with de-centralized control is the robustness of behavior with respect to population size [5].

Experiment 1 Results Summary
(Percent AV Population Reaching Target)

Parameter	Doorway	Hallway	Zigzag
Population –Lo	0.16	0.18	0.20
Population – Hi	0.16	0.18	0.18

Figure 4: Experiment 1 Results for Population

Figure 3: Opera Problem in MASON

Experiment 2 Results Summary
(Number AVs reaching target at time step t)

Parameter	Doorway	Hallway	Zigzag
t	1000	1500	2500
Sensor Range – Lo	11	12	20
Sensor Range -- Hi	9	7	9
Opening Width -- Lo	9	8.5	15
Opening Width – Hi	11	10.5	14
Social Structure – Off	5	7.5	13
Social Structure -- On	16	11.5	15.5

Figure 5: Experiment 2 Results

Summarized results for Experiment 2 are shown in Figure 5. Experiment 2 allowed AVs to survive collision. Given enough time, all AVs will reach the target, so results present a snapshot of those reaching the target at time step t. The time step at which the snap shot occurred was increased for the more complex scenarios to account for the fact that even successful strategies require more time to execute in complex environments. This of course raises the issue of appropriate time step selection. These time steps were selected as the point in each simulation when the rate of change of AVs arriving at the target was the same. The only parameter showing truly consistent behavior across the three environments was the increased sensor range. This implies that, even though reactive agents do not require global knowledge by definition, providing them with too little (or too local) information may not allow successful behaviors to transfer to new situations. There appears to be a threshold at which the degree of information provided to the AVs allows for more robust behaviors. The results for Experiment 3 demonstrated that the effects of social rules increased overall swarm performance, no definite pattern of robustness in behaviors across environments could be confirmed.

CONCLUSIONS AND FUTURE WORK

This section summarizes our results and places them in context of current and future work. We stress that these results are preliminary and require more experimentation by ourselves and others before one can develop a framework of parameters that provide robust behaviors given specific agent capabilities and goals.

Our first result is that the effects of population variation in the swarm were minimal throughout the scenarios. This is a confirmation of the perception that reactive multi-agent systems with de-centralized control architectures are robust in this regard. While cases may be constructed where population extremes affect performance, our data indicates that the performance of multi-agent reactive systems are robust with respect to population size.

Our second result is that increased sensor range afforded better transference of successful swarm behavior than did lower sensor range. This implies that, while reactive agents operating cooperatively in a multi-agent environment do not require global knowledge, the system behavior may not be robust if agents are provided with too little knowledge. System designers may need to be aware that systems designed to provide "just enough" information to agents while maintaining performance in a test environment may show degraded performance more readily than systems that provide more information with no immediate increase in performance in the test environment.

These results require more investigation within the immediate problem domain and other multi-agent systems. Our immediate future work is to extend these experiments to more complicated terrains within MASON, and then to extend to other simulation domains. Longer term goals include exploring these parameters to study the transference of awareness and decision making from simulated to realized robotic agents.

ACKNOWLEDGMENTS

This research is being performed by IST and UCF as part of a larger research effort with Intelligent Systems Technology, Inc. This effort is funded through a DARPA SBIR contract.

REFERENCES

1. Ferber, J., 1999, *Multi-Agent Systems, An Introduction to distributed Artificial Intelligence*, Addison-Wesley, New York

2. Ossowski, S., 1999, "Co-ordination in artificial intelligent societies: social structure and Its implications for autonomous problem-solving agents", in *Lecture Notes on Artificial Intelligence*, Springer, Berlin.

3. Arkin, R. C., 1998, *Behavior-Based Robotics*, MIT Press, Cambridge, Massachusetts

4. Xu, X., Wang, X., He, H., 2003, "A Self-Learning Reactive Navigation Method for Mobile robots", *Proceedings of theSecond International Conference on Machine Learning and Cybenetics*,

5. Tomlinson, B., B. Blumberg and B. Rhodes. 2000. "How Is an Agent Like a Wolf?: Dominance and Submission in Multi-Agent Systems." *International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA '2000),* Wollongong, Australia

6. Goldberg, D, Cicirello, V., Dias, M., Simmons, R., Smith, S., Stentz, A. 2003, "Market-based multi-robot planning in a distributed layered architecture", in *Proceedings from the 2003 International Workshop on Multi-Robot Systems*, Washington, D. C.

7. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., 2004, "MASON: A New Multi-Agent Simulation Toolkit. 2004", *Proceedings of the 2004 SwarmFest Workshop.*