Three papers were chosen for study in the field of natural computing with a common theme of systems of communicating agents, or artificial social behavior. The papers are [Gigliotta2014], [Hoverd2014], and [Massaro2014]. These papers are listed in the References section.

# [Gigliotta2014]

## Summary

### Introduction

[Gigliotta2014] used an evolutionary algorithm to develop optimal role differentiation behavior in a group of robots. They say that role differentiation is an issue which needs resolution when a homogeneous group is to perform a collaborative task. Collaborative, being defined as consisting of different activities performed by specialized individuals. Collaborative, as opposed to coordinated, cooperative, or collective behaviors. Coordinated behavior consists of the group of agents organizing their locations or behavior based on that of the other agents. Cooperative behavior means the agents are interacting so to perform a task demanding more than any individual agent’s abilities. Lastly, collective behavior is when groups of agents make choices on the basis of the outcome for the group.

One might propose using an inhomogeneous group of robots if different roles are required. This does not work in cases where the optimal agent operation parameters are not known. In cases where an evolutionary algorithm is used to search for optimal parameters, multiple agent types necessitates altruism; to find the optimal overall behavior, different agent types must evolve in such a way to improve each other’s fitness, not reflected in their own fitness which is driving their evolution. This problem is avoided if homogeneous robots can differentiate their roles. As this work involves robots differentiating between leaders and non-leaders, and fitness being based on the level of distinction between the leader and other robots, it represents something of an impulse-response test on role-differentiation systems.

### Methods

The robots’ were placed in a virtual square arena and allowed to interact. Leadership was established via infrared signal communication. Robots’ signals ranged from 0 to 1. The robots behavior consisted of how its two motors respond to non-leader, leader, and self infrared communication signals and how its communication output responded to leader and self communication signals. All incoming communication signals had random noise in the range -0.05 to 0.05 added to it. This behavior was encoded in a neural network, the free parameters of which were represented by evolvable genes.

The fitness measure for the evolutionary algorithm is the sum of the differences between the leader, having the largest output signal, and each other robot’s signal, averaged over the number of robots and lifecycles per trial. That is,

(1)

Where C is the total number of cycles for each individual, 40 trials times 1000 cycles per trial for 40000. N is the number of robots in the group, 10. Max is the signal value of the current leader, and O­i is the signal value of robot i.

The initial population consisted of 100 genotypes, each of which was tested for 40 trials each of 1000 lifecycles. The 20 best individuals of each generation are selected for asexual reproduction by making 5 copies of each. Variation is introduced by having 2% of their genes’ bits replaced with random values. They used the best evolved individual resulting from 10 replications of the evolutionary process for subsequent result generation.

### Results

To verify the robustness of their evolutionary method’s solution with respect to scale, they ran 25 trials, each of 3000 lifecycles, for groups of 2, 4, 6, 8, 10, and 12 robots embodying the solution genotype. Average numbers of leaders throughout the trials, and at the ends of trials for the various group sizes were computed. Leaders were defined as robots with outputs higher than 0.5. Groups typically settled on 1 leader eventually, regardless of size. However, as size became too large, this took longer to occur as the arena became crowded. This experiment was also performed with groups of 20, 30, 40, and 50 agents, with standard 150mm communication ranges, and extended 400mm ranges. Despite increasing arena sizes to accommodate these larger groups, performance degraded (average number of leaders increased) with group size. The result of extending the robots’ communication ranges was a mitigation of the group-size related performance

Communicative behavior was investigated by placing two robots facing each other, and running the trial. Each robot’s communication signal value was recorded throughout the lifecycles of this test. The result was that both robots progressively decrease their output until the symmetry breaks at a bifurcation point at 0.5. After this, the output difference becomes amplified and the leader is decided. Without the random noise added to each robot’s perception of the other robots’ outputs, the symmetry is not broken and both robots’ outputs continue to drop.

Non-communicative (motor) behavior was investigated with two tests. In the first, the average speed, and number of robots within communication range were compared between leaders and non-leaders over 100 tests each lasting 3000 cycles with groups of 10 robots. The leaders’ average speeds and connectivities were slightly, but significantly higher than those of non-leaders’. In the second, the average number of leaders throughout and after 10 trials of 3000 cycles each, done on two groups. One group had the usual, best-evolved controlled, and the other had the same with all motor outputs fixed at 0. Being deprived of evolved motor behavior substantially increased the average number of leaders to emerge.

To investigate the importance of interaction topology in this role-differentiation problem, they repeated the evolutionary process with the robots not allowed to move. Robots were initially placed in random locations in the arena subject to the condition that none of them are isolated. The fittest individual typically emerged within 50 generations, with fitnesses around 0.8. Evolution converged more quickly and more consistently than for the standard experiment. Solutions evolved from fixed topology did not perform as well. More leaders were present on average throughout tests, and groups did not converge on a single leader.

## Critique

Despite the number of factors involved in a system of robots establishing their respective roles through communication, [Gigliotta2014] devised a straightforward experiment for isolating and investigating such systems’ properties. Role differentiation was investigated in perhaps its simplest possible rendition, with the goal being for one leader to emerge from a homogeneous group. Behavior consisted of motion controlled by two wheels, and communication in which the robots could either attenuate or escalate their signals. Perhaps the only setup simpler than this, still allowing role determination, was also investigated, wherein the robots could not move. By reducing the experimental setup to such basic components, [Gigliotta2014] performed what might be thought of as a unit-impulse test on a role-differentiating robot system. As such, their design might be used to investigate other attributes of a role-differentiation robot system. Perhaps the robot population could be randomly assigned an ‘affection matrix’, representing the robots mutual feelings, and affecting the results of communication.

The intention of the word “allocation” in the paper’s title might have been captured better by “differentiation”, or “establishment”. “Allocation” seems to imply imposition or dispensation of the roles by a central authority. Their experiment was closer to nature than this. The robots “differentiated” their roles amongst themselves. In the spirit of natural inspiration, one could say that they “established” their roles.

Despite the effectiveness of the genes produced as solutions to the problem of evolving optimal robot attributes, the evolutionary algorithm did not necessarily generate them consistently. They discarded all replications in which the robots did not properly differentiate themselves. This is not a problem within the scope of their work; there is no reason more trials cannot be run to obtain good robot functionality, for good overall fitness. Looking to the future, if real groups of robots are to be deployed into an environment, and left to experiment with different operation parameters, possibly establishing different roles to achieve collaborative behavior, performance must be guaranteed. Algorithms which generate only genes which produce effective phenotypes must be researched. Running evolutionary algorithms with physical robots to search for an optimal behavior pattern with respect to some task would take much longer than using simulated robots. Should the technology progress to the point of applicability, the potential uses are numerous. Anywhere humans have collaborated to perform some task, such as mines, construction, and manufacturing, might stand to benefit from such technology.

# [Hoverd2014]

## Summary

### Introduction

Direct agent-agent communication methods result in deadlock when there is a circular request chain for communication resources among agents. For guaranteed deadlock prevention in a client-server type system, the graph of client-server interactions must be acyclic. This imposes strict constraints on simulations. Such acyclic graphs do not arise in many naturally inspired simulations; one expects many if not all of the agents to be able to interact. Barrier synchronization is described as an unnatural solution to the deadlocking problem. Such solutions might work for certain applications, however barrier synchronization implies a collective coordination. A scripted collective coordination, rather than an emergent one.

To address issues of computational expense and potential deadlock arising from direct agent-agent interaction in complex agent-based systems, [Hoverd2014] proposes the environment orientation approach. Environment orientation is a more natural approach to complex agent-based systems simulation. It reflects reality in that information is passed between agents via the environment. Rather than each agent send messages reminiscent of action-at-a-distance to each other agent requiring that information, each agent passes their external state to the environment. Other agents can perceive other external states subject to environmental modifications or constraints, such as distance or line of sight. Agents repeatedly update their internal state based on their current internal state and the relevant environmental information, followed by their external state and behavior based on their new internal state.

### Methods

To demonstrate that emergent properties observed in conventional direct communication simulations, [Hoverd2014] simulated Reynold’s boids with the usual rules. The information required for individual cohesion, alignment, and separation behavior is conveyed between agents via the ‘visual’ environment. Objects of FlockingNeighborhood class were formed and an object of FlockingAgent class associated with one if the FlockingNeighborhood has at least 10 associated FlockingAgents for at least 20 time steps, and if at least 80% of the FlockingAgents of the FlockingNeighborhood were the same for those steps. To initialize the simulation, 300 boids (FlockingAgent objects) were placed randomly in a circle of 500 unit radius in a virtual, unbounded 2D plane.

They investigated extending simulations to ones containing multiple environments. In this context, multiple environments means agents have access to multiple types of information through each type’s environment. To do this, they simulated Reynold’s boids again, with ‘visual’ information in the ProximityEnvironment, and ‘auditory’ information in the SpeciesEnvironment. This experiment used multiple species of boids. In addition to the usual rules for flocking behavior, members of a common species were attracted to each other by auditory song information.

Finally, to investigate the claim that environment orientation readily allows for agent-environment interaction, they run a hill climbing simulation. Agents were instantiated in a species environment, and a proximity environment, similarly to the multi-environmental boids. Instances of ClimbingAgent class took into account their current ‘altitude’, provided by an object of class LandscapedEnvironment. With this information, ‘visual’ information from nearby agent’s locations shown in their external states by the proximity environment, and ‘species’ information from all ClimbingAgents of a given species provided by the species environment, agents search the landscape for the global maximum.

### Results

From their single-environment boids simulations, [Hoverd2014] collected percentage flocked versus proximity parameter data. Percentage flocked represents the percentage of boids participating in flocking as defined in Methods above. The proximity parameter dictates how close other boids must be for them to be considered as belonging to the same neighborhood. They observed little to no flocking for proximity values less than 25 units, but with percentage flocked moving from 90% to 100% as the proximity value was varied from 40 to 70 units.

In their second set of boids simulations, the proximity parameter was varied again for four more tests, with 1, 2, 3, and 4 agents per species. The number of agents per species is the number of boids per group which attract each other with auditory information regardless of distance. Percentage flocked generally increased with the number of agents per species, and proximity as before. There was, however, no general trend in percentage flocked for proximity values 30 and smaller, across numbers of agents per species. A clear improvement is seen going from proximity values of 35 to 70 for all species sizes.

The success of their hill climbing simulation was assessed on the basis of the percentage of agents over 90% the global maximum value. For this experiment, as for the multi-environmental boids, a proximity parameter and species size were varied from 1 to 6 agents. As one might expect, the percentage of agents above 90% of the global maximum’s increased with both proximity and number of agents per species. With simulations having 4, 5, and 6 agents per species, percentages of agents above the 90% global maximum were roughly 50%, 60%, and 70% for proximity values from 20 to 100. These percentages approached 100% for proximity values of 150.

## Critique

[Hoverd2014]’s work presents an apparent cornucopia of applicability because of the universality of the concept of ‘environment’ to complex agent system simulations. In fact, [Gigliotta2014]’s simulated role-differentiation robots could have incorporated environmental orientation. Although it is difficult to say whether that would yield performance improvements, or merely produce a contrived system, complicating what was an approximation of direct agent-agent communication. It is difficult to predict the utility of environmental orientation for any application from this paper alone. Although they computed performance metrics for their various boids simulations, they did not compare environment orientation’s performance to direct agent-agent communication’s performance.

It is possible that the authors opted to not report agent-agent communication’s performance for fear that it detract from their results. Despite the performance enhancements described in terms of complexity reduction from O(n2), and deadlock prevention, the nature of the boids simulation favored global agent awareness. Assuming the flocking simulation is typically performed with each boid capable of seeing all other boids, agent-agent information provides the boids with information about the entire population, improving flocking behavior, as assessed by [Hoverd2014]. Whereas in the environment orientation approach, where there were multiple small species within the population exhibiting global mutual awareness. As the number of individuals per species increased, it approached the limit of complete global agent awareness, presumably used by the traditional method. As flocking performance improved with the number of individuals per species, it should be optimal using direct agent-agent communication between all agents.

Granted, the performance of their method might not offer improvements in terms of the emergent behavior itself produced. Their method’s superiority to the status quo could have been illustrated by examining computations performed, or machine time required, for simulation for the two methods. This would be more consistent with their discussion of the complexity benefits of their methods, whereas emergent behavior, although interesting, does not relate directly to the claimed benefits of environment orientation.

Despite the work’s quantitative shortcomings, it is conceptually rich. In addition to the expected performance benefits with regard to complexity and deadlock prevention, environment orientation captures something close to the heart of conscious experience by virtue of its natural inspiration and correspondence. As they discuss, agents do not induce states directly in one another; all interaction is mediated by the environment. Environment orientation apparently produces agents which have fallen to the realms human experience. Agents cannot know other agents directly, all information propagation is modulated by the environment. To the end of natural correspondence, one might use agents with no exact representation of their environment at all. Agents’ behavior would be determined entirely by an internal image, filtered and modified by perception and cognition systems, either predefined or evolved.

# [Massaro2014]

## Summary

### Introduction

Several metrics of interest for comparing social network models to real ones include the clustering coefficient, the characteristic length, division in communities and the distribution of the network’s nodes’ connectivity degrees. [Barrat2004]’s definition of clustering is used in this paper. The clustering coefficient is meant to reflect the local group cohesiveness of the network. It is the average each node’s local clustering coefficient, ci:

(2)

The axy terms are the elements of the network’s connectivity matrix, and the inner summation’s coefficient normalizes the local clustering coefficients, accounting for differences in connectivities of the individual nodes. The result is that the local and global clustering coefficients are in the range [0, 1].

Previous network growing methods form realistically structured graphs by relating the probability of new nodes forming connections to existing nodes to the connectivity of candidate neighbors. Although it produces graphs with high clustering, low characteristic lengths, strong division in communities, and variability of degree distribution its workings do not represent those of actual social network growth.

Therefore [Massaro2014] proposes a cognitive-inspired approach to network generation. They claim that this approach is suitable for producing networks whose graphs are comparable to those of actual and theoretical social network graphs. To illustrate this, they describe three models, one of each for generating regular, random, and scale-free networks. They go on to describe two models for producing more sophisticated networks, hierarchical, and random and scale-free networks with community structure. Finally, they describe a model of theirs for creating real world networks.

### Basic network models

The method they report for producing regular graphs, specifically 1D lattices with periodic boundary conditions is as follows. A small, fully-connected graph of m + 1 nodes is initialized as a locus of nucleation. As nodes are added, they are connected to the m closest existing nodes. Having reached the prescribed network size, the original nodes are connected to the newest nodes. The original nodes are missing enough connections to complete those that are still missing from the newest nodes, thus completing the graph by closing a large, symmetric loop in it.

To generate random networks of N nodes, a completely connected graph of m nodes is first initialized. Then, the remaining N – m nodes are added and connected to m randomly chosen nodes of those already existing with uniform probability p. Randomly, subject to the constraint of not creating self-loops or multiple links. This produces graphs whose connectivity degrees follow a normal distribution. The total number of links in the network will then be L = 2mN. The network will have an average connectivity of:

(3)

This method produces networks whose nodes connectivity degrees follow a normal distribution.

Their proposed models for generating scale free networks uses a simple algorithm. Start with m connected nodes to seed the network. Add each of the remaining N – m nodes. As each new node is added, random existing node are selected to be connected to the new node, as long as they are not already connected, until the new node has m connections. This results, again, in a network with L = 2mN links and expected degree of connectivity of 〈k〉 = 2m, and as a scale free network, generates a node connectivity degree distribution of the form P(k) = a∙k-γ. a is some scalar and γ = -3.

### Socially structured network models

[Massaro2014] refers to a method [Massaro2014a] for generating benchmark hierarchical networks against which real networks can be compared. It essentially consists of initializing a symmetric connectivity matrix with no self-loops, and specifying the number of nodes and network subsections there are to be. As nodes are added, they add weights based on cognitive-inspired algorithms meant to infer global network structure from local structure. The result in a connectivity matrix with self-similarity across the length scales of the various subsections.

They then propose a method to produce random networks exhibiting structure similar to real social networks. It models social network community dynamics by assigning different probabilities for a node to form connections with nodes inside, and outside of its community. The number of nodes per community, and number of communities are defined. Then nodes are connected within and without their communities with the corresponding probabilities. This method tends to produce networks with normal distribution of node connectivity degrees, and following power laws.

Their approach to generating scale free networks with social community like structure consists of two parts: generation of scale free communities, and connection of these communities. Each scale free community to exist in the final network is generated using their basic scale free network generation method discussed above. With all the communities created, for each node in the network, with inter-community connection probability: delete a random link, and create a link with a random node in another network.

### Real world network model

Lastly, they discuss their method for producing networks with realistic social structure. Their method is based on the idea that individuals joining networks have a subjective view of the network based on their initial connections within it and the information that gives them access to. When a new node joins the network, it perceives a number of social levels. Each level containing the nodes having a given degree of separation from the new node. The probability of the new node forming connections with existing nodes is reminiscent of the simulated annealing algorithm:

(4)

where a is a normalization constant, β is the (reciprocal) temperature which affects the likelihood of a new node forming connections with deeper social layers, and x is the social layer of the node to which the new one might connect. The observed that the clustering coefficients of these networks tended to decrease with β > 2.5, and increase to a maximum as β varied from 2.5 to 0.1. An opposing trend was observed with the average path length, which decreased for β < 3, and increased for β > 3.

The cognitive-inspired component of this method is introduced as a knowledge vector for each node in the network, forming a knowledge matrix for the network. The element of the knowledge matrix represent the probabilities that one node belongs to the community represented by the other. The knowledge matrix is initialized with the Kroneker delta; each node starts only knowing about itself. The knowledge vectors are updates during a communication phase, then the vectors are used to restructure the network in an elaboration phase. During the communication phase, the knowledge matrix is updated taking into account the last knowledge matrix using a memory coefficient, and the connectivity matrix since members of real networks with large connectivites are expected to have more influence in network dynamics. The network links are updated based on a function of y and the knowledge matrix to simulate human heuristics such as making decisions based on the most important, relevant, piece of information. They found that they could produce networks with characteristic lengths, clustering coefficients, and node degree distributions as two real online social networks.

## Critique

In addition to the quantitative evidence provided for their method’s capability to produce realistic networks, attributes more easily observed than quantified are also in correspondence. As [Massaro2014] discusses, the notion of a ‘community’ in a social network is poorly defined. The boundaries for similarity of connections, or clustering, or whatever simple metric is used to define communities, can be arbitrarily drawn. This is apparent in the method they describe for generating random and scale-free networks with community structure. The number of communities is a predefined integer, and connections are drawn between them. Their proposed method for producing real world networks offers an improvement over this binary treatment of communities.

[Massaro2014]’s method produces realistic community structure by taking into consideration the degrees of separation between incoming and existing nodes. These varying degrees of separation result in the fuzzy nature of community structure in real networks. By allowing connections between varying degrees of separation, rather than community versus non-community, and imposing the realistic factor making more distant connections less likely, [Massaro2014] presents qualitative and conceptual improvements to existing network generation methods, in addition to the quantitative results.

By extending their basic method for realistic network generation to assign knowledge vectors to each node, [Massaro2014] creates a framework for the cognitive inspiration referred to in their title. The knowledge vectors are described as represent nodes’ probabilities of knowing about other nodes’ communities. Further extension of the nodes’ knowledge vectors, and the network’s resulting knowledge matrix, might be used to simulate more sophisticated social behavior. As proposed for [Gigliotta2014]’s role-differentiation robots, [Massaro2014]’s nodes might be given affection vectors. Affection vectors would represent how the nodes feel towards one another and modify the nature of their communication and interaction. For example, they could be calculated on the basis of similarity or complementarity of some arbitrarily assigned traits.

# References

[Barrat2004] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, “The architecture of complex weighted networks”, PNAS 2004; 101(11):3747-3752.

**[Gigliotta2014]O. Gigliotta, M. Mirolli, and S. Nolfi, “Communication based dynamic role allocation in a group of homogeneous robots”, Natural Computing 2014; 13:391-402.**

**[Hoverd2014] T. Hoverd, and S. Stepney, “Environment orientation: a structured simulation approach for agent-based complex systems”, Natural Computing 2014; 14:83-97.**

**[Massaro2014]E. Massaro, F. Bagnoli, A. Guazzini, and H. Olsson, “A cognitive-inspired algorithm for growing networks”, Natural Computing 2014; 13:379-390.**

[Massaro2014a] E. Massaro, and F. Bagnoli, “Hierarchical community structure in complex social networks”, Acta Physica Polonica 2014; 7(2):379-393.