The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Inter-Agent Variation Improves Dynamic Decentralized Task Allocation

Annie S. Wu, Cortney Riggs
Computer Science
University of Central Florida
Orlando, FL 32816-2362
aswu@cs.ucf.edu, cortr@hotmail.com

Abstract

We examine the effects of inter-agent variation on the ability of a decentralized multi-agent system (MAS) to self-organize in response to dynamically changing task demands. In decentralized biological systems, inter-agent variation as minor as noise has been observed to improve a system's ability to redistribute agent resources in response to external stimuli. We compare the performance of two MAS consisting of agents with and without noisy sensors on a cooperative tracking problem and examine the effects of inter-agent variation on agent behaviors and how those behaviors affect system performance. Results show that small variations in how individual agents respond to stimuli can lead to more accurate and stable allocation of agent resources.

Introduction

We investigate the effects of inter-agent variation on the ability of a decentralized multi-agent system (MAS) to self-organize in response to dynamically changing task demands. In decentralized systems, the absence of a central controller means that agents must make decisions on what task to take on independently. In systems where communication between agents is minimal or scaling is necessary, achieving and maintaining an appropriate number of agents on each task can be challenging. Inter-agent variation refers to small differences in how individual agents sense and respond to external stimuli. Biological studies on social insect societies find that small amounts of inter-agent variation as minor as noise are, not only sufficient, but also necessary to the ability of such systems to self-organize. We examine whether inter-agent variation can be beneficial to self-organization in computational MAS by comparing the ability of a decentralized MAS with and without inter-agent variation to dynamically allocate an appropriate number of agents to each task of a cooperative tracking problem.

In the decentralized task allocation problem, multiple independent agents must distribute themselves appropriately among one or more tasks. With no central controller and each agent deciding independently what task to take on and when, the challenge is getting the agents to coordinate such that they do not all do the same thing at the same time, but rather distribute appropriately to meet all task demands. In addition to effectively meeting task demands, such systems also need to be efficient. Efficiency means avoiding or minimizing problems such as extreme responses in which too many or too few agents respond and excessive and unnecessary task switching.

Studies on social insect societies find that "behavioural variability among the workers of a colony is increasingly regarded as fundamental for...division of labour" (Jeanson and Weidenmüller 2014). Genetic noise is the most common source of behavioral variation in biological systems and comparisons of genetically diverse versus similar insect colonies find that genetically diverse colonies can be more stable and efficient (Jones et al. 2004; Weidenmüller 2004). Other sources of behavioral variation include size (Spaethe and Weidenmüller 2002), age (Jeanson and Weidenmüller 2014), and previous experience (Ravary et al. 2007).

The notion that inter-agent variation can affect self-organization in computational systems is not new. A number of studies have shown that inter-agent variation can be helpful for decentralized coordination (Anders et al. 2012; Krieger and Billeter 2000; Riggs and Wu 2012). Ashby (1958) explains the underlying dynamic: because inter-agent variation causes individual agents to respond differently to the same stimuli, as more of the agents in a system vary (number of identical agents decrease), the repertoire of responses that a system as a whole can produce increases. As a result, inter-agent variation allows an MAS as a whole to be able to offer tailored responses to a wider range of problems or scenarios (Ashby 1958; Page 2011).

In this paper, we directly compare the effectiveness and efficiency of two versions of a decentralized MAS – one version with inter-agent variation (M-VAR) and one version without (M-NOVAR) – on solving a dynamic box-pushing problem in which the agents in an MAS collectively push a tracker to follow a moving target. Effectiveness refers to a system's ability to push the tracker to follow a moving target. Efficiency refers to a system's ability to solve the problem with minimal wasted resources or energy. Based on the discussion in (Ashby 1958), we expect the system with inter-agent variation to show gradual entry and departure of agents from the workforce allowing for more system states. We expect the system without inter-agent variation to exhibit more extreme responses and fewer system states. We expect the addition of inter-agent variation result in more efficient systems in which thrashing and task switching are minimized and potentially greater effectiveness at meeting task demands due to more flexible system responses.

System Description

Our testbed consists of a target and a tracker that move in two dimensional space. The movement of the target is specified by one or more user-defined functions. The movement of the tracker is guided by a decentralized MAS. In each timestep, each agent in the MAS can act on one of four tasks in each timestep – *PUSH-NORTH*, *PUSH-EAST*, *PUSH-SOUTH*, or *PUSH-WEST* – or remain idle. The MAS as a whole pushes the tracker in each timestep by generating a movement vector based on the aggregated actions of all of its active agents. Aggregating the actions of multiple agents allows the MAS to push the tracker more precisely than simply north, east, south, or west. The goal of the MAS is to allocate an appropriate number of agents on each task in each timestep to allow the tracker to accurately follow the target.

In each timestep t, the target has a position, a velocity, and a heading. The velocity and heading, denoted by the vector δ_t, are dynamic and may change from one timestep to the next depending on the target's movement function. The position is an ordered pair that is updated in each timestep based on the δ_t for that timestep. We assume that the target and tracker begin each run at the same initial position and that the tracker will not move until the target moves first. Thus, the target will always be at least one step ahead of the tracker in its movements.

Our MAS consists of an "ensemble" (Goldberg and Chen 2001) of n agents, $a_0, a_1, ..., a_n$. In each timestep, each agent, a_i, in the MAS estimates the target's movement vector from the previous timestep δ_{t-1} as $\delta^{'}_{i,t}$. Agent a_i uses $\delta^{'}_{i,t}$ to decide which task, if any, it should take on in the current timestep. The estimate, $\delta^{'}_{i,t}$, is a vector consisting of directional components to the north, east, south, and west. We define a *candidate direction* as any direction in $\delta^{'}_{i,t}$ that has a positive value. Agent a_i chooses a task in timestep t based on the candidate directions available in $\delta^{'}_{i,t}$. If $\delta^{'}_{i,t}$ has a single candidate direction, agent a_i will choose the task that pushes in that direction. If $\delta^{'}_{i,t}$ has more than one candidate direction, agent a_i will choose randomly from among the tasks of the available directions. If $\delta^{'}_{i,t}$ has no candidate directions, agent a_i will remain idle. For example, if the target is moving in the two o'clock direction, an agent will randomly choose between the *PUSH-NORTH* and *PUSH-EAST* tasks. All agents may choose any of the four possible tasks. Not all agents are required to act in every timestep.

Each agent that chooses to take on a pushing task will contribute a movement amount of c towards its chosen direction. In each timestep, the c values from all active agents are summed to form a single movement vector that is applied to the tracker in that timestep. This movement vector is then subtracted from the estimated position change in each direction of the vector. In this way, the estimated position change retains information from previous time steps if previous summed contributions do not accurately match the previous position changes. In other words, dead reckoning error can occur. Because agent contributions are summed, the maximum distance that our control system can move in a single timestep is $\Delta = n \times c$, and this distance can only be reached if all agents choose the same direction of movement. The MAS can keep up with the target movement only when $\Delta \geq \sum_{d \in \delta_t} d$.

M-VAR and M-NOVAR differ in how $\delta^{'}_{i,t}$ is estimated from δ_{t-1}. In M-NOVAR, all agents sense target movement identically and perfectly and, thus, $\delta^{'}_{i,t} = \delta_{t-1}$. In M-VAR, the difference between the estimated $\delta^{'}_{i,t}$ and actual δ_{t-1} values of the target movement is the manifestation of *inter-agent variation*. Each agent, a_i, senses the estimated position change in a direction, $d \in \delta_{t-1}$, with some amount of error, $\epsilon_{i,d}$. These $\epsilon_{i,d}$ values are randomly assigned to each agent and are distributed with a Gaussian distribution with a mean of $\mu = 0$ and variance of $\sigma = 0.2$. Thus, for each direction $d \in \delta_{t-1}$ agent a_i calcuates an estimate $d^{'} \in \delta^{'}_{i,t}$ using $d^{'} = d + \epsilon_{i,d}$. Agents use the perceived values $d^{'} \in \delta^{'}_{i,t}$, not the actual values $d \in \delta_{t-1}$, to determine their direction preference.

We use a simulation testbed for this study because a simulation environment provides greater capabilities for monitoring and recording the details of when and how agents act in response to task stimuli. The detailed data that we are able to capture in simulation allows us to better evaluate the impact and effects of inter-agent variation on the ability of an MAS to self-organize.

Experimental Methods

We compare the performance of M-VAR and M-NOVAR on tracking a target using four different movement functions. The four problems tested are:

- Circular movement: The target starts at the twelve o'clock position and moves clockwise.
- Figure eight movement: The target starts at the center of the shape and moves clockwise around the top loop and counter clockwise around the bottom loop.
- Square movement: The target starts at the center of the top of the square and moves clockwise.
- Random movement: The target wanders randomly in a two dimensional space.

The first three paths (circle, figure eight, and square) consist of 360 timesteps each. The random path experiments run for 500 timesteps. Unless otherwise noted, $n = 100$ and $c = 0.01$. For the M-VAR, $\mu = 0.0$ and $\sigma = 0.2$. Ten runs are performed on each movement function.

We evaluate the effectiveness of each system by examining how accurately the MAS pushes the tracker to follow the target. This is measured in terms of the average distance between the tracker and target throughout a run and the relative distances traveled.

We evaluate the efficiency of each system by looking at how efficiently the agents, which are the resources of the MAS, are being used. An efficient system uses resources judiciously and minimizes wasted energy and effort. We will

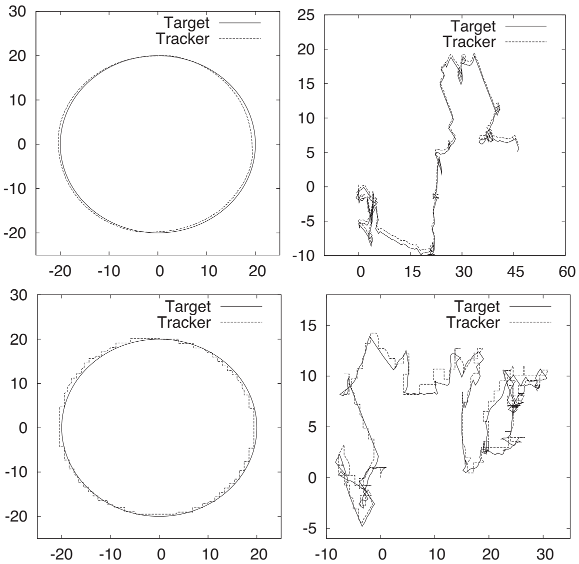

Figure 1: Performance with (top row, M-VAR) and without (bottom row, M-NOVAR) inter-agent variation.

	M-VAR	M-NOVAR
Circle	0.601 (0.030)	0.842 (0.037)
Figure 8	0.411 (0.013)	0.538 (0.026)
Square	0.606 (0.013)	0.706 (0.037)
Random	0.585 (0.028)	0.741 (0.031)

Table 1: Distance between target and tracker averaged over the entire run (and 95% confidence interval) for example runs from Figure 1.

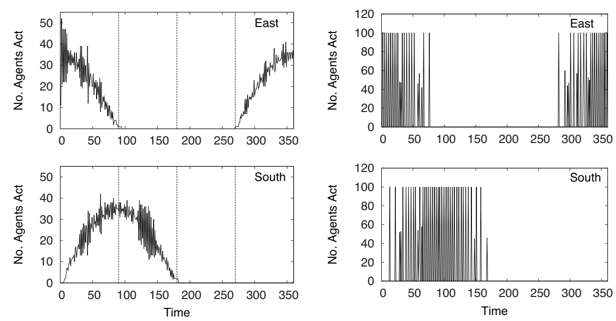

Figure 2: Number of agents that act in each timestep for the circle tracking problem using M-VAR (left column) and M-NOVAR (right column).

monitor the number of agents assigned to each task; the system is efficient if sufficient but not too many agents are assigned to each task. We will also examine the number of times agents switch tasks. For problems where there is a start up cost when starting new tasks, minimal task switching is more efficient.

Results

Figure 1 shows example runs of M-VAR (top row) and M-NOVAR (bottom row) on the circle and random problems. M-VAR consistently exhibits noticeably better performance than M-NOVAR on all tested problems. The M-VAR tracker's paths are smoother and more closely aligned with the corresponding target's paths, indicating that the number of agents allocated to each pushing task varies over time in response to changing task demands. The M-NOVAR tracker's paths are step-like and show more deviations from the corrsponding target's paths, indicating that multiple agents are simultaneously choosing to push in the same direction, and then simultaneously switching to a new task. As a result, the tracker's path exhibits noticeable deviations as agents over-react in one direction then over-correct in another. Table 1 compares the average distance from the tracker to the target averaged over all timesteps of a run and the corresponding 95% confidence intervals. M-VAR distances are significantly shorter than M-NOVAR distances on all four problems.

An examination of the number of agents acting on each task in each timestep supports the above conclusions. Figure 2 shows the number of agents from M-VAR (left) and M-NOVAR (right) that select *PUSH-EAST* and *PUSH-SOUTH* in each timestep for the circular tracking problem. The target and tracker both start the circle problem at twelve o'clock. At timestep zero, the majority of agents in M-VAR select to *PUSH-EAST*. As the target moves around the first quadrant of the circle to three o'clock, M-VAR agents gradually stop choosing the *PUSH-EAST* task and gradually start choosing the *PUSH-SOUTH* task. This gradual departure and entry continues until the target and tracker return to twelve o'clock and the majority of the agents are once again on the *PUSH-EAST* task. M-NOVAR is far less flexible than M-VAR. Although the agents in M-NOVAR are choosing the correct tasks for each period of the run, M-NOVAR agents mostly act in concert and are unable to distribute themselves among tasks based on task demand. Instead, the entire team appears to jump back and forth between tasks, over-responding to one task then another, behavior that explains the stair step trajectories in Figure 1.

In order to compare the efficiency of M-VAR and M-NOVAR, we monitor the number of times agents switch tasks during a run. In many problems, switching from one task to another can entail energy, time, and other start-up costs. In addition, if agents can learn from experience, keeping agents on tasks for which they have previous experience can result in better system performance. As a result, it is often desirable to minimize task switching in MAS. Table 2 gives the average number of task switches that a single agent undergoes over an entire run for each of the four problems and the corresponding 95% confidence intervals. M-VAR agents undergo significantly fewer (up to 5 times less) task switches than M-NOVAR agents. Interestingly, the

	M-VAR	M-NOVAR
Circle	47.72 (7.235)	271.19 (0.260)
Figure 8	60.36 (6.780)	263.23 (0.271)
Square	145.75 (15.151)	357.00 (0.000)
Random	157.44 (9.789)	377.35 (0.547)

Table 2: Average number of task switches that an agent undergoes during a single run (and 95% confidence interval).

	Target Length	M-VAR Length (Excess)	M-NOVAR Length (Excess)
Circle	125.31	127.23 (1.53%)	153.80 (22.73%)
Figure 8	121.45	124.20 (2.27%)	154.22 (26.98%)
Square	179.50	179.33 (-0.10%)	179.00 (-0.28%)
Random	195.76	187.69 (-3.66%)	234.67 (19.87%)

Table 3: Length of target and tracker paths for the example runs from Figure 1 and the percent excess of each tracker path over the corresponding target path.

confidence interval for M-NOVAR is significantly less than the confidence interval for M-VAR. We believe this is due to the fact that most agents in M-NOVAR are making the same decisions and switching to the same tasks at the same time.

The last performance metric that we compare is the actual distances travelled by the target and tracker. Table 3 gives the length of the target and tracker paths for the example runs shown in Figure 1. In general, trackers guided by M-VAR travel paths with lengths that are shorter and closer to the target path length than trackers guided M-NOVAR. The one exception in which M-NOVAR travels a shorter path than M-VAR is the square trajectory which is a very simple trajectory that only travels in one direction at a time and, hence, should be easy for a system like M-NOVAR where agents tend to act in concert. Interestingly, in that problem, both trackers actually travel slightly shorter distances than the target. Similarly, M-VAR's tracker also travels a shorter path than the target in the random movement problem. We speculate that, because M-VAR's tracker is trailing the target and because of the haphazard movements of the random function, there may be timesteps in which the target moves toward the tracker. On the other hand, M-NOVAR's tendency to allocate all agents to the same task at the same time amplifies the difficulty of following a target making random abrupt movements.

Conclusions

In this paper, we compare the performance of a decentralized MAS with and without inter-agent variation, respectively called M-VAR and M-NOVAR, on allocating agents to solve a cooperative tracking problem. We test both systems on a simulation problem in which the agents in a decentralized MAV cooperatively guide a tracker to follow a moving target by individually selecting one of four directions to push. The actions of all agents are aggregated in each timestep to generate tracker movement. Inter-agent variation is added to M-VAR in the form of sensor noise.

Although both systems are able to track a moving target, our data indicate that M-VAR is significantly more effective and efficient than M-NOVAR at solving the problem. M-VAR pushes the tracker along a path that is more accurately aligned with the target's path because M-VAR is better able to distribute agents among the four pushing tasks appropriately to address changing task demands. M-VAR follows the target more closely than M-NOVAR and typically travels a shorter path than M-NOVAR. M-VAR is also more stable than M-NOVAR, with agents switching tasks more than five times less often in M-VAR than M-NOVAR.

The idea that noise manifested as inter-agent variation can improve system wide behavior in decentralized systems potentially very interesting. Physical components of autonomous multi-agent systems are commonly subject to noise, e.g. calibration errors, wear and tear, dead reckoning errors, general physical imperfections, etc. A great deal of energy and research is often dedicated to minimizing and overcoming such noise. Imagine if, instead, these natural physical occurrences may be harnessed to improve multi-agent system coordination and behavior.

References

Anders, G.; Hinrichs, C.; Siefert, F.; Behrmann, P.; Reif, W.; and Sonnenschein, M. 2012. On the influence of inter-agent variation on multi-agent algorithms solving a dynamic task allocation problem under uncertainty. In *Proc. IEEE 6th Int'l Conf. Self-Adaptive & Self-Organizing Systems*, 29–38.

Ashby, W. R. 1958. Requisite variety and its implications for the control of complex systems. *Cybernetica* 1(2):83–99.

Goldberg, K., and Chen, B. 2001. Collaborative control of robot motion: robustness to error. In *Proc IEEE/RSJ Int'l Conf Intelligent Robots and Systems*, 655–660.

Jeanson, R., and Weidenmüller, A. 2014. Interindividual variability in social insects – proximate causes and ultimate consequences. *Biological Reviews* 89:671–687.

Jones, J. C.; Myerscough, M. R.; Graham, S.; and Oldroyd, B. P. 2004. Honey bee nest thermoregulation: Diversity promotes stability. *Science* 305(5682):402–404.

Krieger, M. J. B., and Billeter, J.-B. 2000. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems* 30:65–84.

Page, S. E. 2011. *Diversity and Complexity*. Princeton University Press.

Ravary, F.; Lecoutey, E.; Kaminski, G.; Chaline, N.; and Jaisson, P. 2007. Individual experience alone can generate lasting division of labor in ants. *Current Biology* 17:1–5.

Riggs, C., and Wu, A. S. 2012. Variation as an element in multi-agent control for target tracking. In *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, 834–841.

Spaethe, J., and Weidenmüller, A. 2002. Size variation and foraging rate in bumblebees ($Bombus\ terrestris$). *Insectes Sociaux* 49(2):142–146.

Weidenmüller, A. 2004. The control of nest climate in bumblebee ($Bombus\ terrestris$) colonies: Interindividual variability and self reinforcement in fanning response. *Behavioral Ecology* 15:120–128.