

Task Allocation in Distributed Artificial Intelligence using Boids Model

Muhammad Radwan, Amr Badr, Ibrahim Farag
Computer Science Department, Faculty of computers and Information,
Cairo University, Egypt

ABSTRACT

Decentralized multi-agent approach is a promising research field particularly in the area of performance improvement by handling task allocation and communication time. Some recent research has focused on developing the learning process to be better suited for specific problems; other efforts had proven that a generalized solution is better off especially when there is no global controller. This paper presents a better suited multimembered evolution strategy to agent reasoning with an improved method of pre-assigning initial values to agents. We show through computer experiments that agents using the presented method reach a stable state in a faster pace than other multi-agent systems, although after a stable state is reached the improvement -we are presenting- effect will be a little limited until the system reaches an unstable state again.

Keywords

Decentralized multi-agent approach, MAS, evolution strategy

1. INTRODUCTION

An intelligent agent is a sophisticated entity that act autonomously by perceive information from the environment using sensors ,decide to take action through reasoning process to perform its role and lastly act upon its environment through actuators . Intelligent agents have been used to provide solutions in almost every real life domain from aviation, supply-chain management, land reclamation, industrial processing to learning assistants. Agent oriented systems depended on a single agent until the 80s when it was realized that individual agents are becoming too complex to function as expected to. So, divide and conquer approach was implemented leading to distributed AI ^[1], which was shortly evolved to the modern Multi-Agent Systems.

The rise of the Multi-Agent approach opened the door to more efficient agent based technology as each agent can adapt to

solve a particular problem in the system or specifically while managing communication at the same time to share information or current time results, exchange roles and avoid interdependence problems. Distributing the system over multiple platforms and hardware enhances the robustness and reliability significantly. In addition Multi-Agent approach improves computational efficiency by distributing work load. It also increases extensibility, maintainability, flexibility and reuse as the agents are more decentralized.

Since then there were efforts to improve Multi-Agent performance and design specifically in Negotiation, Problem Solving, Learning, Organizational Control, Coordination and industry driven research. Negotiation research is more concerned about communication between agents ^[2]. Problem Solving field uses cooperative distributed agents to solve specific problems, where the main target is creating suitable algorithms that facilitate agent results reusability across the system for better processing time ^[3]. Learning research topic is concerned about processing the agent history to facilitate taking better optimized decisions in different situations inside the system environment ^[4]. Organizational Control involves redesigning agent topology and relation to better adapt other problems or larger systems ^[5]. Coordination area of research deals with coordinate agents' activities and agents' synchronization when dealing with system environment and resources ^[6].

Research efforts often handle more than one point of the previously mentioned research areas, as all are closely related to each other together with industry driven research as urban traffic simulation, wireless sensors negotiation or medical risk assessment ^[7]. Unfortunately, many of agent based systems nowadays are still depending on a derivative of the individual agent paradigm that implements a controller agent or even a group of agents each controlling a certain criteria which undermines Multi-Agent Systems basic advantages.

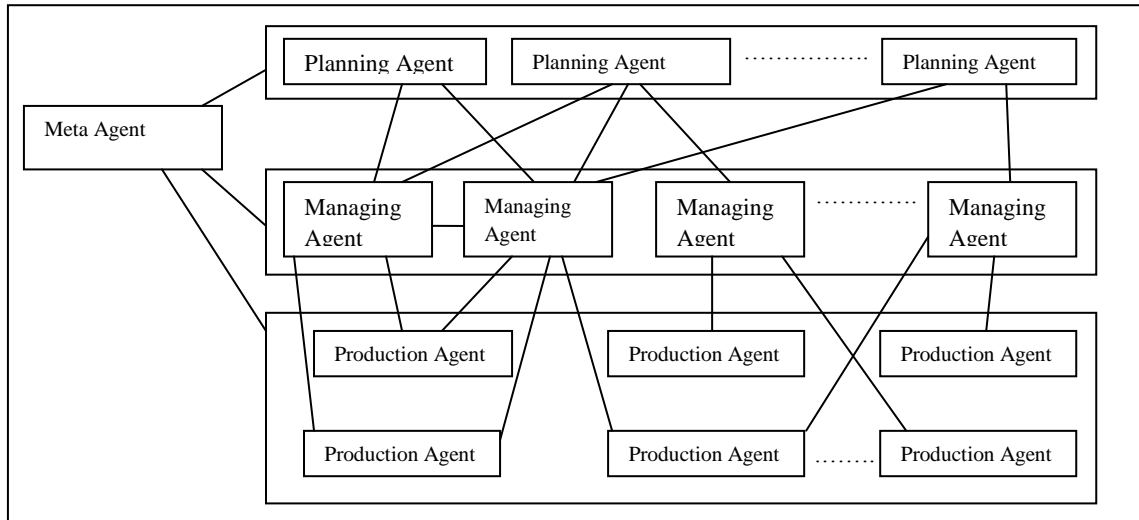


Fig.1. Contract Net's project agent can perform two different functions: management or production agent (contractor). Management agent monitors task execution and processes task results at multiple levels so a hierarchy of management agents can exist, production agent executes tasks, project planning agent is a special management agent that assumes the position of a top manager agent and Meta agent roles is monitoring and maintaining the content of agents regardless of their roles.

Centrally controlled pre-designed agents inherited the same bottlenecks and critical time failure of individual single agent specially the "single point of failure" that were the first motivation towards Multi-Agent research [8]. In this paradigm each working agent has a manager or more to response to and then the managing agent resend data to be processed to other agent so the managing agent acts as a proxy that controls which agent get what information or view of the system. In many cases the managing agents have other manager or managers to manage as well. As one agent or more in the agent management hierarchy exhibit even a slight latency the whole system will be delayed or worse (failure).

Availability and robustness are the main Advantages of a Multi-Agent Approach due to decentralization and distribution of resources among multiple agents with no single pre-designed agent. The reason behind that design is biologically inspired models where a random agent can be assigned a de facto controller rule due to a temporary implicit leadership caused by incident of getting the best input from the environment .The biologically inspired model is rooted in the herd mechanism of leadership where few informed members of the herd lead most of the others as each member of the herd base its decision on the movement of other neighboring member with no explicit leader and each member is simply following the rest and affecting the group decision one way or another. In another words, the agents are distributed randomly across the system where each agent gets a random position so the input it perceive from the environment will be random as well. Due to the variation of location and data perceived , the decisions and calculations will be different so at least one agent will be getting favorable results ahead of his peers then by communicating with them it will implicitly direct all other agents to more optimal state closer to achieving the system objective.



Fig.2. flock of birds in a uniform stable movement

2. System main features

When discussing multi-agent approach or dealing with research areas, the question arise which scenario are we dealing with and under what conditions? So, we are presenting a system that is generalized as much as possible. Luckily decentralized multi-agent approach is one of the most promising areas of research and better suited to these conditions as any success in a decentralized generalized environment can be transformed and implemented in other specific situations that are primarily of relative centralized design as medical applications, supply chain analysis... etc

There is no better model for decentralized MAS than the biological model that inspired multi-agent approach itself, so the research used "boids implementation" as one of the most popular biological models. From that well known system the research started investigating the effects of the improvements we are presenting.

The research first intention was to improve performance multi-agent systems through decreasing communication time and the negotiation procedure as the drawback that hindered the biologically driven model was the higher processing time in comparison to the centralized approach. So, the presented system took two directions.

2.1 Improvement of the initial positioning

In a centralized driven approach, one or more agents are per-configured to manage the rest of the agents where every agent system is uniquely designed to offer the best and most efficient solution to a specific problem. The drawback was the same point, each implementation is unique that old efforts are lost in the process and no code reusability is achieved in most cases as discussed earlier. That what made a reliable decentralized approach more desirable than before, in the case of centralized approach when a single controller agent fail the entire system will not recover, a decentralized system will not fail as other agents can take leadership when others fail.

The system we are proposing at this point is trying to simulate the main advantage of the centralized implementation that was its unique design for each problem through jumping directly from the initial point to a stable state. And it tries to achieve this point by analyze the system interactions and outputs each time frame or number of hopes and test what if this state is the first initial value and by having multiple candidate values, it can easily analyzed to select an optimal start position using any well known optimizer as genetic algorithms or similar technologies . Although the achieved state is better suited initial values to the agent, it will not as effective as the pre-designed centralized approach but will make decentralized systems in a fairer match with the centralized systems in the case of performance and that was the motivation to the second point.

2.2 Mutation of the agent reasoning

Most of the mainstream decentralized multi-agent systems depend on simple reasoning and more interaction with other agents to achieve a stable status. This approach costs more communication time and less processing time at each agent. In a real life situation this is not usually the case as communication time may be more costly than taking a bit more thinking at each agent location. An example of that case is robotics implementations where the communication and interaction can be a lot costly than having a bit more processing at each node. To handle this issue we are presenting another biologically inspired trend from another relative field. A derivative of multimembered evolution strategy was implemented at the agent level that the agents are now taking more responsible actions and achieving its goal using less interaction with the environment with no overload on the processing time at each agent level.

3. Related Works

The field has attracted the interests of many researchers over the time. Chih-Han Yu presented an algorithm that allowed every agent to perform individually without supervision from a controller and as agents reach favorable results it will send its data to other agents in the system ^[9]. As the agents are identical in implementation, some agents reach same or similar results. By using this fact, agents calculate probabilities of results propagation through the system and choosing the best values that reflect collective decision and implicit leadership.

Mihail , Tuyls and Nowe research article presented a learning algorithm that decreases latency with limited power without having a centralized control. In a conventional adhoc network all agents equally listen and sleep for a pre-defined amount of time. The results showed that agent learning outperform adhoc network with higher energy efficiency ^[10]. Agent learning leap at one agent is communicated to a local view of other agents that will benefit of neighborhood and in turn will transfer the new knowledge to other agents in the system. So, one agent learning can guide the entire system to more efficient performance.

Kagan Tumer and John Lawson investigated multi-agent coordination approach to distribute job scheduling across heterogeneous servers. Load balancing would have solved this problem but on the condition of having a centralized controller ^[11]. Instead, the global evaluation function was eliminated and replaced by objective functions at agent level. The agents are decentralized but each agent or group of agents is unique to reflect local goals for each. After testing using moderate problems the improvement was marginal over load balance but fortunately that was with the absence of a centralized controller agent. But, in a complex situation the outperformance was clear.

Although the paper presented by Satoshi Takahashi and TokuroMatsuo intention was to solve a specific e-commerce trading scheme problem, the solution the paper presented was a much more generalized model where they eliminated seller agents and buyer agents and used trader agents instead ^[12]. The new trader agent bases its selection on probabilities of success with other agents. The probabilities used are based on evaluation of other trader agents.

4. Multi-agent System Implementation

The well known Boids model was designed in the 80s to emulate flocks of bird, schools of fish and similar movements in the nature. The boids is a totally decentralized system where each boid representing a bird take its decision individually with only local view of the other birds in the herd. With only three decisions to make while moving avoiding collision, steering and rotating in the main area of the other birds (boids). So, there is no centralized agent or a hierarchy of agent management where every movement is running smoothly by local decisions only.

The boids implementation is based on choosing random values for the agents' data as the usual in decentralized multi-agent based models but we analyzed this approach as previously discussed and exchanged that with a preprocessed values while in the same time perform an evolution strategy on agent level that is discussed latter in more details

4.1 Framework used

The implementation is based on Janus- project kernel and associate libraries. Janus is an enterprise-ready open-source multi-agent platform fully implemented in Java 1.6 with the most up to date features and requirements that missed many older agent-based modeling tools. The paper also used Janus-project basic implementation of boids as a ground on which we constructed the presented model and to have a standard implementation to compare with as well.

4.2 Boid agent objective

The boids artificial life simulation follows three rules at the agent reasoning.

4.2.1 Separation

To avoid collision with other boids through keeping a distance from nearby flock mates. The crowding boids steers away from each other by a force relative to the distance in between them.

4.2.2 Alignment

Through emulating the orientation of the local flock mates by keeping track of the direction other boids are heading and simply following them while affecting the overall flock direction as the boid faces other effects as the separation even from other population group of other flock or cohesion with nearby boids of other local flock.

4.2.3 Cohesion

That can be achieved by keeping around the center of the entire flock or the local view of the flock and the longer the distance from the center the higher the force that will be applied.

The basic rules described above have been improving over the time and more rules have been added. As, setting a direction goal simulating flock migration, limiting the force that can be applied to the boids to keep a speed limit relative to the one in real life or even perching as the birds tend to land and stay on the ground from time to time. But we followed the basic model only as these improvements are handling the artificial life simulation point which is not the point of this research.

4.3 Agent learning

The modification we are presenting is a form of multimembered evolution strategy. That's applying mutation to the neighboring boids in each individual boid local view. In the multimembered evolution strategy parents are selected from the population regarding a global view but in the boids case there is no global view. Luckily, the local view simplified the selection process as less boids will be dealt with. As all neighboring boids must effect the boid decision, all must be selected to be parents to the new generation.

After parent the selection, multimembered evolution strategy assumes two ways of evolution (m+1) and (m,l) also known as plus strategy and comma strategy, respectively. In the plus strategy both the parent and offspring enters the selection process while in the comma strategy only the offspring does [13]. In the our system, although only the offspring take part in the mutation (evolution) this will happen for the local environment not the total system and for an optimized number of generations other than reassessing the system each hope as the assessment would have moved the implementation into a centralized system.

4.4 System evaluation criteria

In the supposed mechanism the overall objective function evaluate the effort until the system reaches a stable (stable state is the state where each boid population is aligned together in an organized formation with small rate of collisions) .This can be implemented in 2 ways. The first by measuring the time to a stable state, the other by measuring the number of actions that agents took until the stable state is reached.

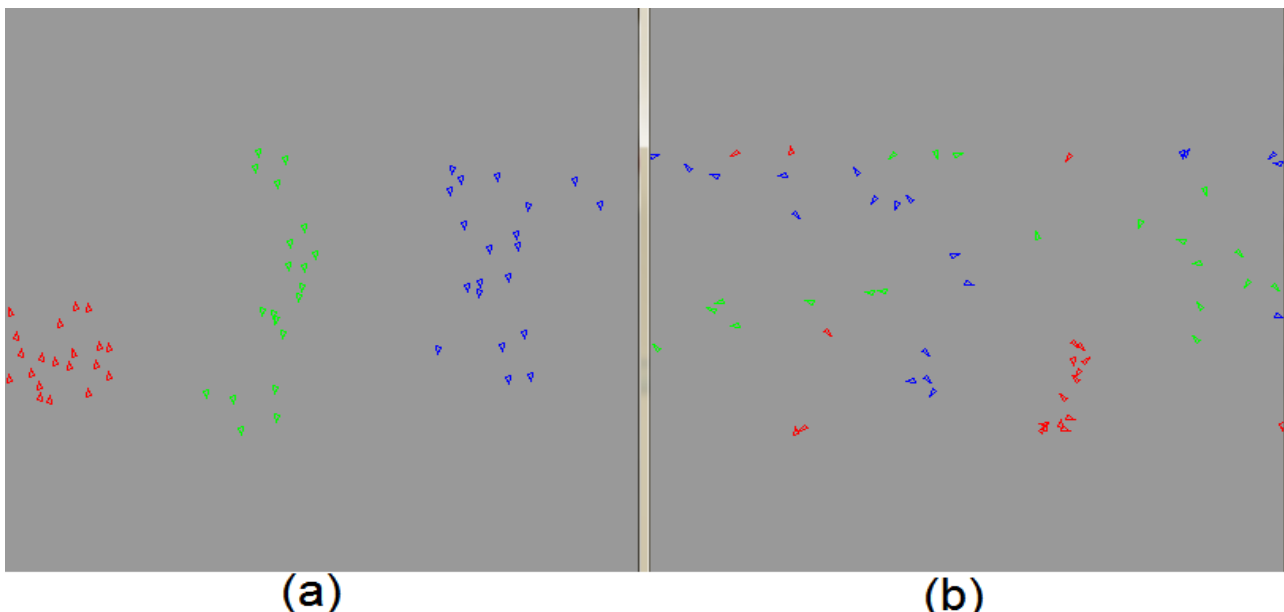


Fig.3. (a) stable boid flocks where each boid agent of the 3 present flocks is moving in the same direction and with similar velocity as other boids of the same flock while keeping around its the flock center. (b) Unstable boid flocks as most boids local view is narrow and reflecting small clusters of boids or even a single boid of a given population flying alone with independent direction, speed and away from flock center. This state usually arise as boids are randomly scattered across the environment or an obstacle faced the flocks that can even be a forth flock or a flock that stabilized in an opposite direction to the other.

For each population or a single population of boids, here in this case a single population group, we add all the distances between every boid agent and the center of the flock.

$$\min \sum_{i=0}^n \sqrt{(xi - x)^2 + (yi - y)^2}$$

Where n is the number of boids in a given population, x is the mean x location, y is the mean y location

For the entire boid simulator in case there is more than one population group, the system will simply add all the populations' evaluation results.

$$\min \sum_{p=0}^{pn} \sum_{i=0}^n \sqrt{(xi - x)^2 + (yi - y)^2}$$

Where n is the number of boids in a given population, x is the mean x location, y is the mean y location

The evaluation function handles the three rules of the boids. The cohesion is measured by keeping a minimum distance

from the central position of all the boids in the entire flock not just the local as the case a single agent deal with. If there is a collision or separation the boid agents will steer away from each other maximizing the distance between them which in turn will maximize the evaluation function value away from the optimal spread out. And in the case of alignment, if the boids are not in the same heading direction and relative speed, they will go distantly apart from each other away from the evaluation criteria as well.

5. Experimental results

As the research took two directions each will be evaluated individually.

5.1 The initial pre-assigned states

The optimal states that were used improved the evaluation function results significantly in a short time as expected .So, the experimentation intended to measure its effect not on the short period but on the intermediate period and the long term.

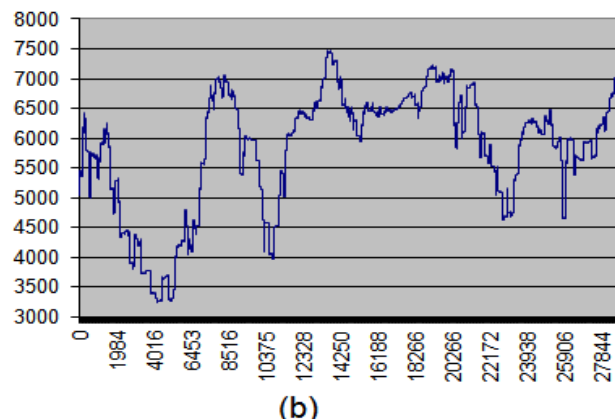
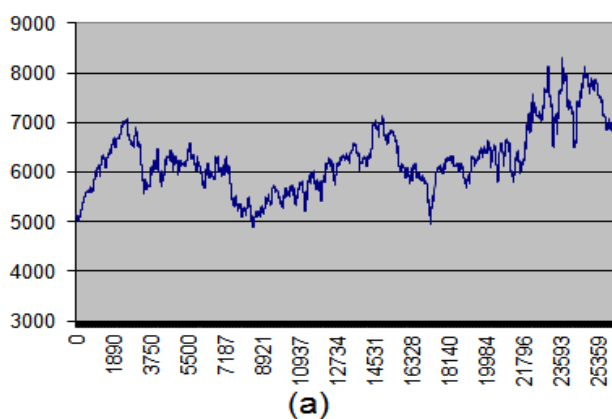


Fig.4. the x-axis represents the time and the y-axis represents the evaluation function values (a) without using pre-assigned initial values (b) with using pre-assigned initial values

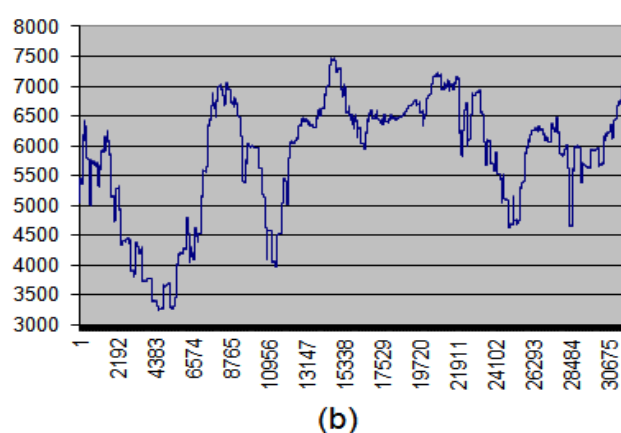
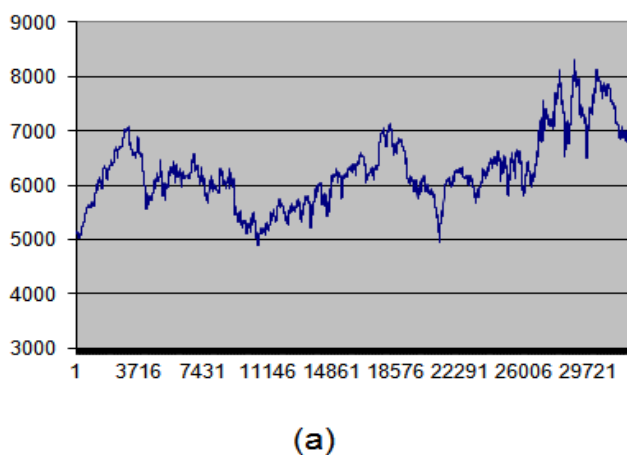


Fig.5. the x-axis represents the hop count and the y-axis represents evaluation function values (a) without using pre-assigned initial values (b) with using pre-assigned initial values

By testing we found that although it did improve the performance for most cases this effect fades with the time. "most cases" term was used because a decentralized multi agent system is stochastic by nature where the optimal start point may not be the best option in some cases as the time passes and the boid agents deal with the environment and

other agents the results cannot be expected in a certain way . At this point the randomly assigned initial states may yield better result after the system run for an intermediate time period.

5.2 Mutation of the agent reasoning

As the main target of multi-agent systems are real life situations or simulation, the main issue becomes of improved thinking process before interaction rather than a rapidly calculated interactions. In other words, agents with better

thinking and less affecting of the environment will get an improved results than agents that interact and act upon basic biologically inspired conditional rules, so our research has moved in the direction of an Intelligent Biological system.

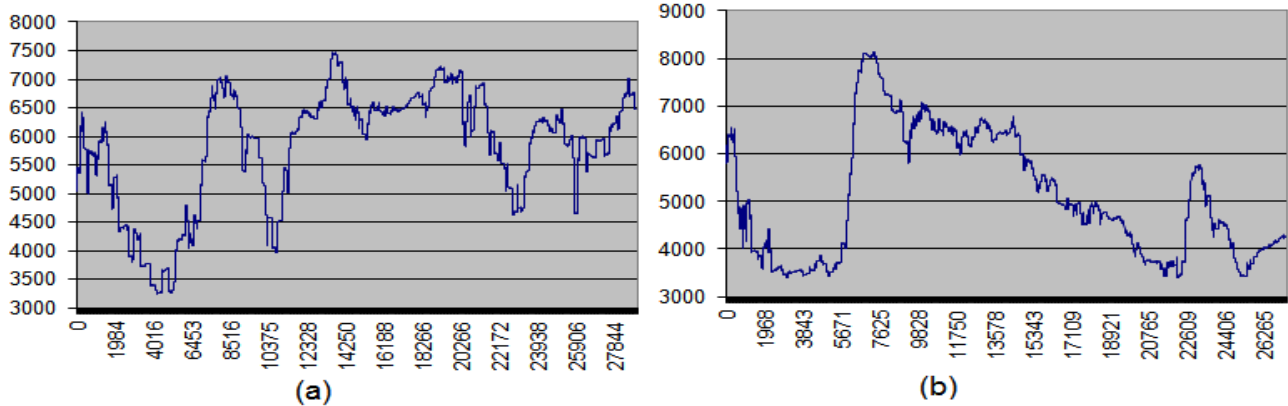


Fig.6. the x-axis represents the time and the y-axis represents the evaluation function values (a) without using mutation (b) with using mutation at agent reasoning level

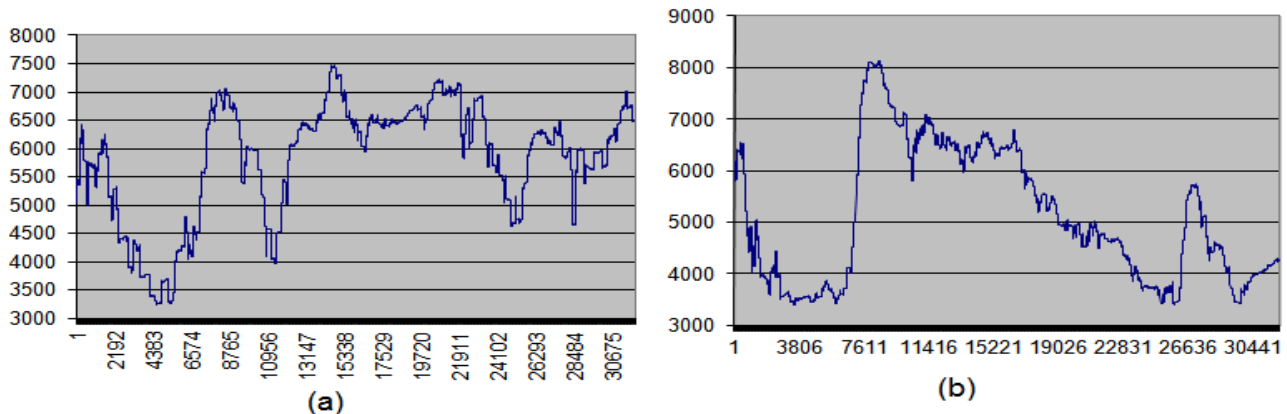


Fig.7. the x-axis represents the hop count and the y-axis represents evaluation function values (a) without using mutation (b) with using mutation at agent reasoning level, where its noticeable that (b) has a tendency to escape the high levels of instability (higher y-axis values) faster than the former but while the system is in a stable position the difference become hard to notice

The improvement was more clear when dealing with disperse scattered boids , as the boids of the same population are scattered across the environment while dealing with collisions with other population groups the stable state is achieved faster when using the evolution strategy described previously than the normal reasoning. While , after reaching a stable state , there will be little difference between the two mechanisms until the flock or sub-flock faces an obstacle or environmental change that cause it to disperse where the presented agent learning modification will return to a relatively better performance .

6. Conclusion and Future work

The experimentation investigated the effect of the two presented points on improving the decentralized multi agent approach performance. The first point was using pre-defined initial values for the agents rather than randomly assigned values without turning into a centralized approach model, which turned out to improve the performance at the short term while its effect will fade overtime. Because when the system reaches a stable state, the rate of agent failure will decrease sharply and the process of assigning values will be used rarely. The second point was about improving the agent

reasoning by implementing an evolution strategy aiming at optimizing the force affecting the agent. Firstly, It was suspected that the number of hopes (decisions) will improve at the expense of processing time because each agent will take longer time of thinking (more processing time) taking the best decision so the number of times it effect the environment (hops) until it reach stability will decrease. But, the result was that both improved at the same pace where the processing time at agent level did not increase as it was expected. So, with less hope count, same processing time at agent level, the system total performance improved.

Agent learning process research has a lot of potential as many improvements can be tested and implemented. The paper has presented an evolution strategy hybrid with better performance, and this can easily relate to using algorithms inspired by neural networks learning and fuzzy logic that can be researched to further upgrade the performance of agent reasoning.

7. References

- [1] Victor Lesser and Daniel Corkill, History and Accomplishments of the Multi-Agent Systems Lab at the University of Massachusetts Amherst, 2011, pp. 1-5.
- [2] Bo An , Automated Negotiation for Complex Multi-Agent Resource Allocation, (Dissertation Submitted to the Graduate School of the University of Massachusetts Amherst in partial fulfillment of the requirements for the degree of PhD, 2011, pp. 2-15, 138-167, 202.
- [3] Huzafa Zafar, Victor Lesser, Daniel Corkill and Deepak Ganesan, Using Organization Knowledge to Improve Routing Performance in Wireless Multi-Agent Networks, 2008, pp. 8.
- [4] Raphen Becker, Analyzing Myopic Approaches for Multi-Agent Communication, Computational Intelligence, Volume 25, Number 1, Blackwell Publishers, 2009, pp. 44–45.
- [5] Mark Sims , Daniel Corkill and Victor Lesser , Automated Organization Design for Multiagent Systems, Autonomous Agents and Multi-Agent Systems, Springer Netherlands, Vol. 16 (2), 2008, pp.181–183.
- [6] Thomas A. Wagner, Achieving Global Coherence in Multi-Agent Caregiver Systems, 2002, pp. 1-2.
- [7] Thao P. Le, Timothy J. Norman and Wamberto Vasconcelos, Adaptive Negotiation in Managing Wireless Sensor Networks, PRIMA Kolkata, India, November 12-15, 2010, pp. 121,228, 511.
- [8] Wei Ren, Yang Xu, Jinmei Luo and Liying Guo , Influence of Social Networks on Recovering Large Scale Distributed Systems ,PRIMA 2009, LNAI 5925,2009, pp. 579–586.
- [9] Chih-Han, Biologically-Inspired Control for Self-Adaptive Multiagent Systems, PhD thesis, Harvard University, 2010.
- [10] Mihail Mihaylov, Karl Tuyls, and Ann Nowe, Decentralized Learning in Wireless Sensor Networks, LNAI 5924, 2010, pp. 60–73.
- [11] Kagan Tumer and John Lawson, Coordinating Learning Agents for Multiple Resource Job Scheduling, LNAI 5924, 2009, pp. 123–137.
- [12] Satoshi Takahashi and Tokuro Matsuo, A Cooperation Trading Method with Hybrid Traders, PRIMA 2009, LNAI 5925, 2010, pp. 608–615.
- [13] Hans-Georg Beyer and Hans-Paul-Schwefel, Evolution strategies A comprehensive introduction, 2002.