Centralised Autonomic Self-Adaptation in a Foraging Robot Swarm

Liam McGuigan, Roy Sterritt, George Wilkie
School of Computing, Faculty of Computing, Engineering and the Built Environment
Ulster University
Jordanstown, N. Ireland
email: mcguigan-l8@ulster.ac.uk, r.sterritt@ulster.ac.uk, fg.wilkie@ulster.ac.uk

***Abstract*— The use of robotic swarms in domains, such as space exploration, or search and rescue missions, requires that the swarms be self-adaptive in order to adjust to newly acquired data, and react to unforeseen events. Research on swarm self-adaptation tends to focus on the adaptation of individual agents, however taking a top-down approach can allow for the use of knowledge that is only apparent considering the swarm as a whole. This research makes use of a centralised Autonomic Manager to modify the behaviour of a simulated swarm of foraging robots by adjusting the range over which individual robots broadcast help requests. The swarm is able to learn its own size and the size of the test area, and use that information to guide its decision making, showing the potential for a future decentralised approach. First, the swarm is tasked with recognising the initial situation. Secondly, the swarm must respond to two events which alter the scenario parameters, namely the destruction of a proportion of the swarm, and a change in effective communication range. Performance of the swarm using an Autonomic Manager is compared against that using a fixed broadcast range suited to the initial circumstances. The results show that the swarm can recognise the initial situation and select a suitable broadcast range. It is also capable of recognising the events that occur, but the effectiveness of its response depends on additional parameters in the simulation.**

Keywords- Swarm robotics; Self-adaptation; Autonomic Computing; Simulation.

I. INTRODUCTION

A swarm of robots, in which the aggregate behaviour of many relatively simple individuals combines to create a more complex set of behaviours [1], can have applications in areas, such as mine clearance [2], search & rescue [3] and space exploration [4][5]. A robot swarm can reduce the demands on any single robot, may accomplish the task more quickly, and can be deployed where sending humans is too dangerous, difficult, or costly.

The ability to self-adapt, that is to adjust behaviour in response to newly acquired information without the need for external guidance, is a requirement of a robotic swarm [6]. Unforeseen events may occur that require adjustment, and factors, such as distance and time, may restrict the ability for a human operator to act successfully. Self-adaptation can be applied to the swarm in a variety of ways [7], including the development of emergent behaviours [8], evolutionary systems [9] and swarm-level decision making [10].

Autonomic Computing concepts [11][12] can be used for swarm-self adaptation. At the swarm level, an Autonomic Manager (AM) employing a control loop, such as the Monitor, Analyse, Plan, Execute system described by [11] can be used to allow the swarm to assess the current situation and take any action necessary. This may be implemented in a centralised manner, with individual robots communicating with a central command unit, or decentralised with each robot using its own control loop in order to modify its own behaviour in response to shared information and experience.

The objective of this work is to explore the potential for using swarm-level self-adaptation in a swarm of robots to improve performance in a foraging task, specifically the time it takes the swarm to complete the task which may often be an application priority, such as in search & rescue.

Robot swarms are typically decentralised in nature [2], however here a centralized approach is used as an initial exploratory stage to determine if an AM provides any benefits, with the work of decentralisation to follow this research. As such, the centralised AM here is limited to analysing aggregate data and adjusting parameters, rather than taking a more active role in coordinating the swarm.

The AM aims to achieve performance improvement through modification of the range at which individual robots communicate with neighbouring robots for assistance. The swarm is tasked with deciding the appropriate communication range, and then two unforeseen events are introduced. The first, robot destruction, tests the swarm's ability to react to the sudden change in swarm size, such as a loss of robots in a search & rescue task due to the hazardous environment. The second, a change to communications quality, represents a situation where the ability of the robots to communicate with each other may be hampered by a change in environmental conditions.

The rest of this paper is structured as follows. Section II discusses related work in swarm-level adaptation. Section III describes the simulation used and swarm task, the implementation of autonomic behaviour in the swarm, and the scenarios tested. Section IV reports these results and explores the implications. Section V concludes the paper with a summary, and future research directions.

II. RELATED WORK

The location where adaptation is applied to a swarm is important when considering the intended goal. Much of the research in adaptation focuses on the level of the individual agent, where the resulting swarm performance is affected by the aggregate of these individual behaviours [7]. This level of adaptation can have a dramatic impact on performance, but it is difficult for any single robot to take advantage of information that is only available when viewing the larger

 ISBN: 978-1-61208-854-9

picture, or to make decisions affecting the behaviour of other members of the swarm, such as cooperation or communication.

Adaptation at the swarm level can counter some of these problems. [13] describes an approach to moderating the size of the swarm in order to reduce degraded performance due to congestion. Robots keep track of the conflicts that occur when two robots attempt to occupy the same cell. If the number of conflicts crosses a threshold, virtual pheromones can be deposited at the entrance in order to instruct robots to leave or join the area. Hence, the swarm can adjust its size based on the combination of each robot's collision tracking data.

In [14], a group of unmanned aerial vehicles (UAVs) are patrolling an area defined by a set of cells, with the aim of ensuring that cells are visited often enough during the mission. Individual UAVs decide their next target on the basis of values assigned to the cells by a central system based on UAV visitation. Different strategies for assigning those values are explored, and so the central system becomes an effective behaviour adaptation method for the group.

As discussed in Section I, autonomic concepts may be used for swarm self-adaptation. [15] describes an adaptation pattern in which one robot in the swarm takes on the role of an AM, running a control loop with visibility of the whole system. In the case study presented, the swarm was tasked with exploring an unknown area. Robots communicate their positional and explorational information with the AM, which can direct them to underexplored areas. Recognising that a centralised system may be a bottleneck, a decentralised variant is also used in which the robots share the information with their neighbours. Both approaches perform much better than a basic pheromone-based approach.

A partially distributed approach described in [16] uses a group of UAVs, together with communication base stations taking on the role of AMs, engaged in a search task. If one of the UAVs leaves the active area and loses the communication link, the base stations are able to recognise the failure and reposition themselves in order to retrieve the UAV, while also minimising disruption to the rest of the swarm.

In a previous paper [7], cooperation strategies for swarms were investigated to determine the potential for using an AM to select between them based on the situation. This research builds on that by using a centralised AM is employed to modify the broadcast range parameter to explore how an AM can improve performance over using a fixed strategy.

III. SIMULATION SETUP

The following subsections introduce the simulation setup and describe the specific task the swarm must carry out. This is followed by details of how the autonomic management of the swarm functions, and a description of the test scenarios run.

A. *Simulation and Task Description*

This research employs a time-stepped simulation of a heterogeneous swarm of agents engaged in a variant of a foraging task, as reported in previous work [7].

The simulation creates a world with a rectangular grid of cells, seeded with several items with an associated type, and several robots with corresponding types, as shown in Figure 1. Only one item may be generated on a single cell, however any number of robots may stack. A cell can be considered to represent a much larger area than the footprint of a single robot, leaving plenty of room for multiple robots per cell, thus allowing the simulation to ignore potential collisions. The simulation proceeds in a time-stepped manner – each tick of the simulation, all robots are updated in turn.

Each map is seeded with several items, which have an associated type, and several robots with corresponding types. Robots will initially search for items using a wander behaviour, selecting a random, valid direction each update to move, and moving one cell in that direction. On finding an item, the robot will forage if it matches the type, however if the types differ the robots may broadcast a help message to recruit a suitable robot within range. Foraging is carried out in-situ, rather than returning an item to a base. The process may be considered analogous to applications, such as mine deactivation, analysis of mineral deposits, or environmental clean-up.

The cooperation process is the Help Recruitment strategy as described in the previous work [7] – the robot broadcasts the help message and waits for responses. If multiple responses are received, the nearest robot is selected and assigned the task. While a robot waits for responses or task assignment, it remains stationary until the process is complete.

Communication messages are queued and processed at the end of each simulation tick. First, each message is sent to all robots within range. After all messages are sent, each robot shuffles the list of unread messages, and then processes each in turn – in this way, the simulation can avoid the update order being a factor in the behaviour of the robots. Without shuffling, if a robot was to receive two help requests in a single tick, it would always respond only to the first one.

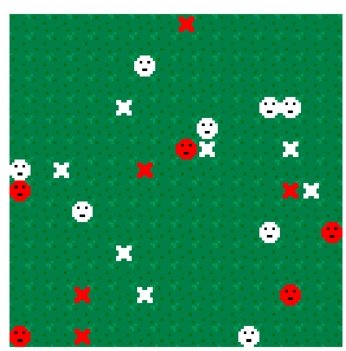


Figure 1. A portion of the world state during a simulation. The colour of a robot (face) or item (cross) indicates its type.

 ISBN: 978-1-61208-854-9

Communications are affected by a global quality setting, which acts as a multiplier on the range of each message sent – a value of 50%, for example, results in a message intended to be sent 10 cells to only reach robots within 5 cells.

A measure of energy expended by each robot is tracked by assigning actions an energy cost. Each robot incurs an upkeep cost of one unit per tick, in addition to the cost of actions taken. Foraging an item costs 1 unit, and movement costs 1 unit per cell moved, and 1.41 units for diagonal movement. Communication cost depends on the maximum range of the broadcast according to the power law stated in (1), where r is the range of the broadcast in cells.

$$\text{cost} = 0.01 \times r^2 \tag{1}$$

Energy is measured in arbitrary units and is designed as a means of exploring the potential impact on a swarm of using increased broadcast ranges.

B. *Autonomic Robots*

Each robot contains an autonomic management component to gather and process information local to the robot, which is then sent to a centralised AM to make swarm-wide decisions, which occurs every 32 simulation ticks. This value was chosen to balance the need to react to situations with the desire to avoid an increase in communications needed to allow for higher AM update rates.

Each robot keeps track of the rectangular region of the map it has so far explored, and sends a synchronisation message to the central AM containing that, and the robot's type. The central AM uses the aggregate data of all robots to estimate the total map size as a rectangle containing all individually explored regions, as well as the swarm size and composition by totalling individual robot types.

Additionally, each robot sends a pulse message with the same period as the central AM message. This pulse is sent to other robots within a fixed range of 8 cells to allow identification of neighbours. Before sending the message to the central AM, the robot calculates the maximum distance from which it received a pulse message from other robots, and sends this to the central AM, which in turn records the maximum distance received by any robot. This can be used together with the known pulse range of 8 cells to detect any possible changes in communications quality.

The central AM uses the information received to determine the best range at which to broadcast help requests, seeking to balance the need for a broadcast to reach a recipient, with the increased energy requirements of broadcasting at higher ranges and the impact on the swarm performance as more robots receive and respond to help requests.

To do this, the swarm uses the map size and swarm composition to calculate the density, δ, of the robots of each type within the world, as in (2),

$$\delta = r / a, \tag{2}$$

where r is the number of robots of a given type, and a is the total area of the map.

The ideal broadcast range was determined by measuring the number of simulation ticks it takes the swarm to complete the task under a selection of broadcast ranges and swarm sizes, and selecting the ranges with the shortest ticks to completion for each size, as shown in Figure 2. Fitting an approximate trend line to the plot leads to an equation for determining the broadcast range based on the lowest density of any given robot type, as in (3).

$$\text{range} = 2.6594 \times \delta_{min}^{-0.46} \tag{3}$$

The ideal range can then be divided by the estimated communications quality in order to determine a suitable range to counter its effects. Finally, the range used is clamped between 1 and 128 before being communicated to the individual robots.

If necessary, the central AM can also decide to halt any attempt at cooperation. If communications quality drops to zero, there is no need to spend time sending messages and waiting for replies when those messages will never arrive.

C. *Test Scenarios*

Three sets of tests were conducted. First, the central AM performance was measured in set of fixed scenarios. Second, the ability of the AM to react to the sudden destruction of a proportion of the robots was tested. The third test tested the AM's ability to react to a change in communications quality.

Each test was carried out with a 128x128 map, seeded with 256 items, equally distributed between two types. Each scenario within a test was run 100 times to obtain a sample, and performance has been measured based on simulation ticks to complete the task. Additionally, the energy cost during the task has been measured.

1) Central AM Performance: To test the hypothesis that the AM is capable of selecting a suitable broadcast range and perform no worse than the best fixed setting, three sets of scenarios were used, consisting of 64, 128 and finally 256 robots, equally distributed between the two item types. Each scenario was run with a set of 9 fixed help recruitment broadcast ranges set at 4, 8, 16, 24, 32, 40, 48, 56 and 64

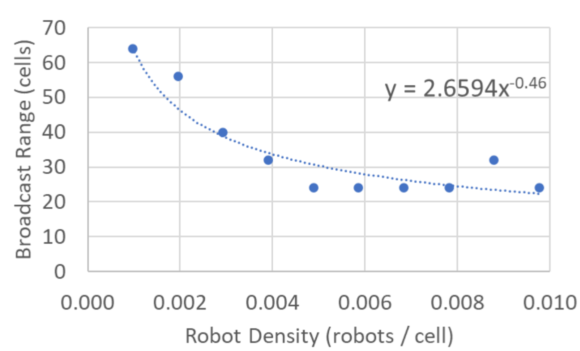


Figure 2. Derivation of the ideal broadcast range function. The points indicate the best performing broadcast range tested for the given density, based on mean simulation ticks to completion.

cells. Finally, each scenario was run with an active autonomic management system to select the best broadcast range.

2) Robot Destruction: To test the central AM's ability to recognise a sudden change in swarm composition, the 256-robot scenario was run with an event scheduled to occur after 300 simulation ticks, in which a given percentage of robots – equally split between the two types – is destroyed. The percentages employed in the test were 25%, 50%, 75% and 90%. Tests were run with the best performing fixed help broadcast range, as identified during the Central AM Performance test above, and then again with the active AM.

3) Communications Quality Change: To test the central AM's ability to recognise a change in the communications quality, the 256-robot scenario was run with an event scheduled to occur after 300 simulation ticks, in which the communication quality changes. The changes employed were 100-25%, 25-100%, 100-0% and 0-100%. As before, tests were run with the best performing fixed broadcast range, and then again with the AM.

IV. RESULTS

The following subsections discuss the results of the three main test scenarios, followed by an overall summary.

A. Central AM Performance

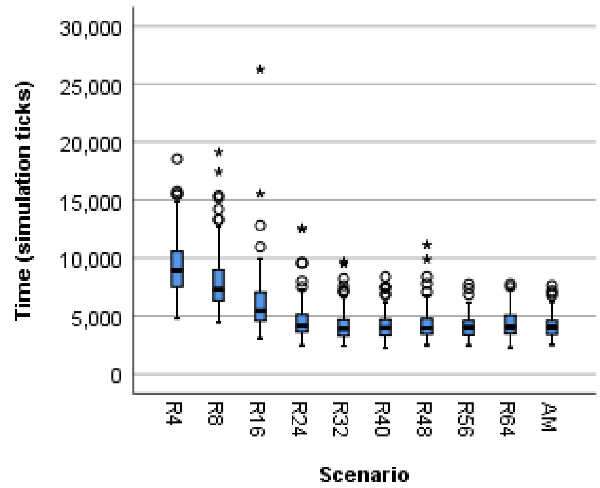
Figure 3 shows the performance of the swarm in each strategy for the three swarm sizes tested, while Figure 4 shows the total energy cost during the test for a swarm of 256 robots.

Independent t-tests were performed between the identified best broadcast range for each swarm size, against the performance of the central AM. The results of this are summarised in Table I.

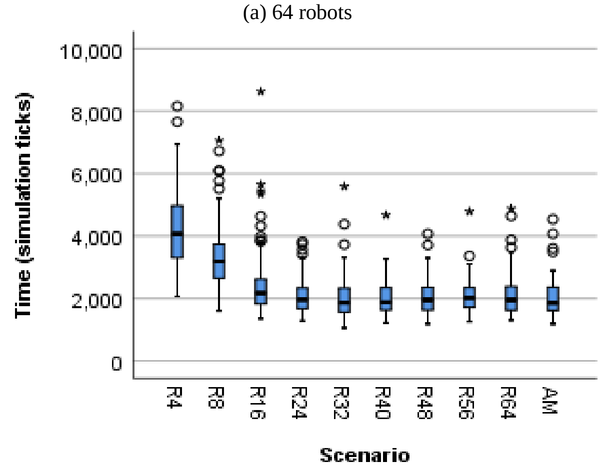
The results show that a swarm using a central AM, making decisions based on the aggregate data collected by individual robots, is capable of selecting an appropriate broadcast range for the Help Recruitment strategy used. It can be seen from the t-test results that there is no statistical difference between the best performing fixed range, and the use of an AM, at a 95% confidence level. This applies for all three swarm sizes tested.

Figure 3 (c) and Figure 4 show that while broadcast ranges of 16 cells and higher show similar performance when measured on completion time alone, the energy demands on the swarm increase when the range grows beyond 24 cells. Therefore, it is not sufficient to set the swarm to operate with a higher broadcast range in order to cover any eventuality, as the swarm would become less efficient.

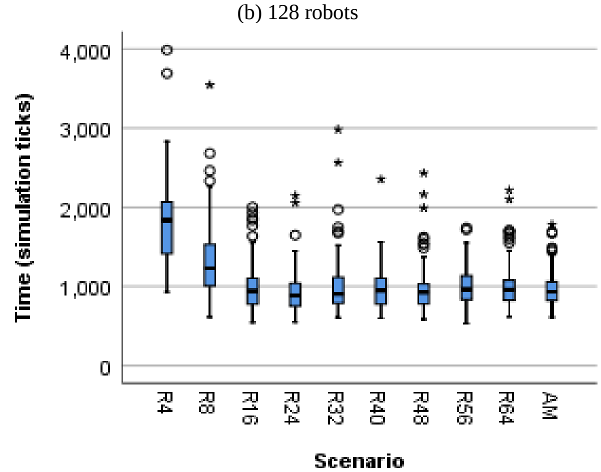
These findings show that in situations where the swarm size and operating area cannot be predicted ahead of time, the ability to determine a suitable broadcast range during the task can prove beneficial to overall swarm performance.



(a) 64 robots



(b) 128 robots



(c) 256 robots

Figure 3. Simulation ticks to complete foraging task for each broadcast range tested, and with an AM, for three different swarm sizes: (a) 64 robots, (b) 128 robots, and (c) 256 robots. Circles and crosses indicate outliers in the data.

TABLE I. CENTRAL AM PERFORMANCE T-TEST RESULTS

Swarm Size	Ideal Range	Fixed Range		AM		Deg. of Freedom	t-statistic	p-value
		Mean	Std. Dev.	Mean	Std. Dev.			
64	56	4151.59	1031.553	4153.54	1044.386	198	-0.013	0.989
128	40	2022.26	564.917	2024.05	596.051	198	-0.022	0.983
256	24	936.90	267.334	983.77	248.969	198	-1.283	0.201

TABLE II. ROBOT DESTRUCTION T-TEST RESULTS

Destroyed Robots / %	Fixed Range		AM		Deg. of Freedom	t-statistic	p-value
	Mean	Std. Dev.	Mean	Std. Dev.			
25	1187.51	342.602	1150.36	303.995	198	0.811	0.418
50	1630.59	617.813	1811.13	606.876	198	-2.085	0.038
75	3664.05	1709.930	3041.41	1226.075	179.516	2.959	0.004
90	11196.18	5458.347	7974.65	2753.011	146.305	5.270	0.000

B. Robot Destruction

Figure 5 shows the performance of the swarm under each test scenario. Independent t-tests were run comparing the fixed range performance with that where the AM is active, and the results are summarised in Table II.

The results here are not so clear cut. In the cases where 75% and 90% of robots are destroyed, the AM's ability to adjust the broadcast range to compensate for the decreased swarm density proves beneficial to the overall swarm performance. In situations where robots may be lost due to hazardous environments, this would prove useful.

However, the t-test results in Table II show that at the 50% level, the AM actually reduces overall performance. This result is surprising given destroying 50% of the robots leads to a remaining swarm of 128 robots, and the results of the Central AM Performance tests show that the AM performs as well as the case with a fixed broadcast range of 24 cells. Further investigation was conducted by running the fixed range and AM tests in this case a further 500 times each. The results of that test show no statistical difference between the two cases, suggesting random chance was responsible for the results in Table II for the 50% destruction test.

C. Communications Quality Change

Figure 6 shows the performance of the swarm and the energy cost for each communications quality change scenario.

Independent t-tests were run comparing the fixed range performance with that where the AM is active, and the results are summarised in Table III. The equivalent tests comparing energy usage are shown in Table IV.

The results show that the AM only improves in both performance and efficiency in the situation where the communications quality drops from 100% to 25%. The AM is able to adjust the broadcast range to compensate for the decreased communications range.

Where quality increases from 25% to 100%, the AM does not show any performance advantage. This is likely

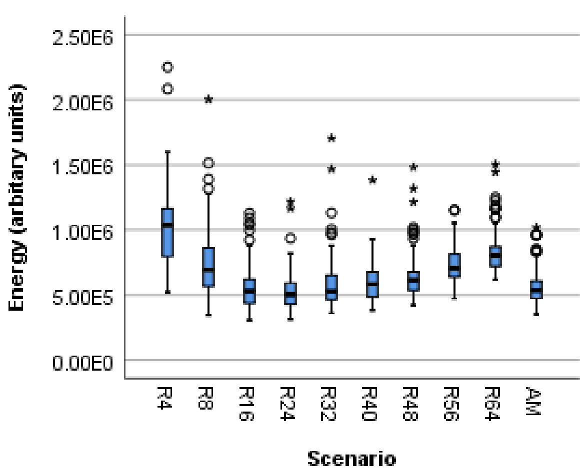


Figure 4. Energy cost for the swarm of 256 robots during the foraging task, for each broadcast range tested, and with an AM. Circles and crosses indicate outliers in the data.

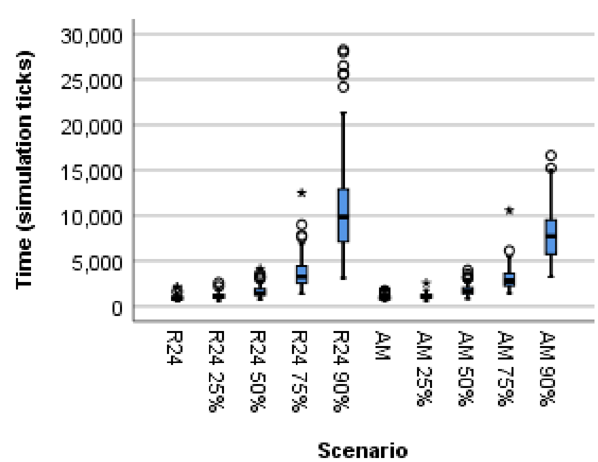


Figure 5. Simulation ticks to complete foraging task for each robot destruction scenario, with an without an AM. Circles and crosses indicate outliers in the data

due to the higher number of items during the earlier stages of the task. As help requests cause robots to stop the search for a while to participate in the recruitment process, higher ranges cause more robots to halt what would otherwise be a fruitful random search. In this scenario, the AM is also less efficient, a consequence of broadcasting at a higher range.

In the cases where the communications quality begins or ends at 0%, no statistical differences can be seen between the AM and a fixed broadcast range. This is likely because at 0% communications quality, no cooperation is possible, and the performance of the swarm is dominated by the random search for items.

D. Summary

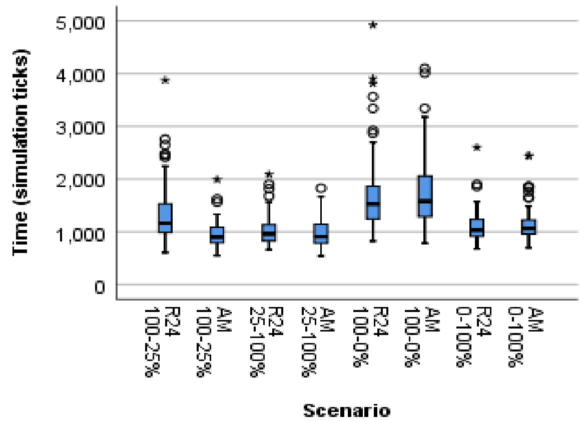
The results above show that the presence of an Autonomic Manager can have benefits for the performance of the swarm, however it is possible for the AM to reduce performance in some circumstances. These situations will require further investigation, and the AM may need to be improved in order to take into account further variables in order to counter their effects. For example, if estimates of the density of items in the world can be made, this could be used to reduce communication range when the density is high, avoiding the interruptions that may lead to the poorer performance in this period.

V. Conclusion and Future Work

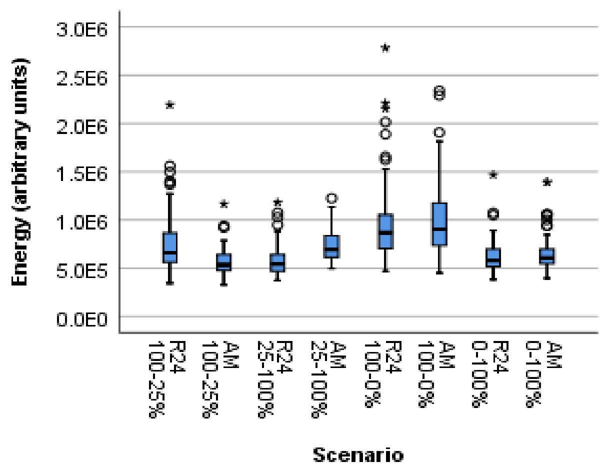
This research used a simulation of a robotic swarm equipped with a centralised Autonomic Manager capable of managing performance through the adjustment of the intra-swarm communication range.

The findings show that an AM is capable of finding an appropriate communication range when given a task where the map size and number of robots in the swarm is not initially known to the AM and must be deduced from information gathered by individual robots.

When a robot destruction event occurs, the AM proves beneficial to the swarm when the robot loss is high, capable



(a) Simulation ticks to complete



(b) Energy cost

Figure 6. Performance of the swarm in the communications quality change scenarios, with and without an AM: (a) simulation ticks to complete, and (b) energy cost. Circles and crosses indicate outliers in the data.

TABLE III. Communications Quality Change T-Test Results - Ticks

Quality Change	Fixed Range		AM		Deg. of Freedom	t-statistic	p-value
	Mean	Std. Dev.	Mean	Std. Dev.			
100 – 25%	1314.86	518.360	944.96	239.680	139.481	6.477	0.000
25 – 100%	1015.27	268.303	980.89	286.952	198	0.875	0.383
100 – 0%	1663.61	706.816	1724.27	625.697	198	-0.643	0.521
0 – 100%	1108.01	276.859	1132.90	302.442	198	-0.607	0.545

TABLE IV. Communications Quality Change T-Test Results - Energy

Quality Change	Fixed Range / 1000		AM / 1000		Deg. of Freedom	t-statistic	p-value
	Mean	Std. Dev.	Mean	Std. Dev.			
100 – 25%	746.49	292.797	563.98	136.668	140.184	5.648	0.000
25 – 100%	574.95	151.535	730.59	163.040	198	-6.993	0.000
100 – 0%	943.32	399.203	986.42	356.575	198	-0.805	0.422
0 – 100%	627.33	156.363	646.28	172.379	198	-0.814	0.417

of completing the task faster than using a fixed broadcast range. No benefit is seen when the robot loss is low.

In the event of a change in communications quality, the AM is capable of improving performance when the quality drops from high to low without dropping out entirely, but not when the quality starts low and increases. This is likely due to the increased item density during the early stages of the task, and it is worth exploring this factor to see how the AM might measure and take item density into account.

It is noted that this work was conducted using a centralised AM that makes global decisions on behalf of the swarm. Such a system introduces problems that have not been replicated in this work, such as the potential for the central AM to be a bottleneck on performance, the presence of a single point of failure, the need for individual robots to maintain that link, and reduced autonomy of any one robot. Future work will include producing a decentralised autonomic layer within the swarm, where individual robots run their own AMs that make decisions based on local data and swarm-level information that can be shared through the regular pulse messages.

Future work may also explore other situations that may affect performance, such as more complex maps containing obstacles, differing distributions of robots, more complexity in the foraging task, on-board batteries that drain and require recharging, and further events that may occur to unexpectedly change the world state.

REFERENCES

[1] H. Hamann, *Swarm robotics: A formal approach*. Springer International Publishing, Cham, Switzerland, 2018.

[2] I. Navarro and F. Matía, "An Introduction to Swarm Robotics", *ISRN Robot.*, vol. 2013, pp. 1–10, 2013.

[3] L. Bayindir, "A review of swarm robotics tasks", *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016.

[4] R. Sterritt, G. Wilkie, C. Saunders, M. D. C. Gama, G. Hawe, and L. McGuigan, "Inspiration for Space 2.0 from Autonomic-ANTS (Autonomous NanoTechnology Swarms) Concept missions", presented at the Reinventing Space Conference, 2019.

[5] A. Farahani, G. Cabri, and E. Nazemi, "Self-* properties in collective adaptive systems", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, Heidelberg, Germany, Sep. 2016, pp. 1309–1314.

[6] G. Beni, "From Swarm Intelligence to Swarm Robotics", in *Swarm Robotics*, Berlin, Heidelberg, 2005, pp. 1–9.

[7] L. McGuigan, C. Saunders, R. Sterritt, and G. Wilkie, "Cooperation Strategies in a Time-Stepped Simulation of Foraging Robots", in *The Twelfth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2020) IARIA*, Oct. 2020, p. 8.

[8] J. Prasetyo, G. De Masi, and E. Ferrante, "Collective decision making in dynamic environments", *Swarm Intell.*, vol. 13, no. 3, pp. 217–243, Dec. 2019.

[9] N. Bredeche, E. Haasdijk, and A. Prieto, "Embodied evolution in collective robotics: A review", *Front. Robot. AI*, vol. 5, p. 12, 2018.

[10] C. Saunders, R. Sterritt, and G. Wilkie, "Collective Communication Strategies for Space Exploration", *J. Br. Interplanet. Soc.*, vol. 72, no. 12, pp. 416–430, 2019.

[11] J. O. Kephart and D. M. Chess, "The vision of autonomic computing", *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[12] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, "Swarm Technology at NASA: Building Resilient Systems", *IT Prof.*, vol. 14, no. 2, pp. 36–42, Mar. 2012.

[13] J. Zelenka, T. Kasanický, and I. Budinská, "A Self-adapting Method for 3D Environment Exploration Inspired by Swarm Behaviour", in *Advances in Service and Industrial Robotics*, Cham, 2018, pp. 493–502.

[14] K. S. Kappel, T. M. Cabreira, J. L. Marins, L. B. de Brisolara, and P. R. Ferreira, "Strategies for Patrolling Missions with Multiple UAVs", *J. Intell. Robot. Syst.*, vol. 99, pp. 499–515, Sep. 2019.

[15] M. Puviani, G. Cabri, and F. Zambonelli, "Agent-based Simulations of Patterns for Self-adaptive Systems":, in *Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, ESEO, Angers, Loire Valley, France, 2014, pp. 190–200.

[16] M. Allison, M. Robinson, and G. Rusin, "An Autonomic Model-Driven Architecture to Support Runtime Adaptation in Swarm Behavior", in *Advances in Information and Communication*, Cham, 2020, pp. 422–437.