Accepted for: Reinventing Space Conference
Paper ID: BIS-RS-2019-06

Collective Communication Strategies for Space Exploration

Catherine Saunders
Ulster University

Roy Sterritt
Ulster University

George Wilkie
Ulster University

17th Reinventing Space Conference

Belfast 12-14 November 2019

BIS-RS-2019-06

Collective Communication Strategies for Space Exploration

Catherine Saunders, Roy Sterritt, George Wilkie
Ulster University
Shore Road, Newtownabbey, Co. Antrim, BT37 0QB;
ce.saunders; r.sterritt; fg.wilkie; @ulster.ac.uk

ABSTRACT

In this paper we compare the efficiency of three co-operation strategies that enable a simulated swarm to collaborate on a searching task. Communication between swarm entities takes the form of decentralized direct peer-to-peer messages. This is in contrast with most swarm research which seeks to mimic the indirect communication observed in nature. A software simulation was created using C#, swarm robots of various colours are placed on a 2D grid and given the task of finding and collecting items that match their colour. If they encounter an item that they cannot collect they send a message to their closest neighbours who then pass it on to their neighbours, robots that match the colour then respond to the help message and move to the item.

The co-operation strategies dictate how far the message will be sent and how many robots are allowed to respond, depending on the strategy the number of responders can be one or many robots. The first strategy permits an unlimited number of swarm robots to respond to a help message. However, the amount of time it takes to find all items can be negatively impacted due to the mass movement of the swarm to this one item. The second strategy limits the number of responders to a select few, the message is not passed on further if within the first message pass there are swarm entities that can respond to the message. The third strategy requires a back and forth dialogue in which only one responder is chosen by the robot sending the help message. The swarm uses Dijkstra's algorithm to find the shortest path when navigating the grid, they are given random coordinates to move to, these are updated when the target is reached. When moving to the target they scan each grid cell for items, if an item is found it is either foraged or a help message is sent to other swarm members. The research seeks to compare the performance metrics of the different strategies. Most notably, it looks at how much overall effort is wasted responding to help messages and whether one strategy is ideal given a certain swarm size. Future research will seek to implement an Autonomic Manager which can reconfigure the swarm composition and communication strategy in real time.

Swarm communication techniques will become more relevant as future space missions progress from a monolithic rover to autonomous swarms. As the numbers of rovers increase it will be more difficult and time consuming to control each with navigational waypoints. The communication lag to and from Earth could prove costly if a rover encounters a hazardous situation or needs to request help from a rover that carries a specific instrument. Being able to act autonomously and collaborate on a task would speed up mission completion time and provide a more robust and adaptable model for exploration.

KEYWORDS: Swarm, Spacecraft, Co-operation, Autonomic Computing, Communication, Robots

1. INTRODUCTION

The future of unmanned space exploration will see the development of swarms of spacecraft that can collaborate with each other and adjust their behaviour to suit the environment they are exploring. They will be designed to be autonomous and autonomic so that they are able to adapt their behaviour to suit changeable mission circumstances. [1] Each swarm spacecraft needs to be autonomous and not controlled by a human or directed via waypoints. Autonomic Computing [2] derives inspiration from biology and seeks to instil more self-management within each spacecraft and the swarm as a whole. An Autonomic Element consists of an Autonomic Manager (AM) and a Managed Component (MC) which sends data back to the AM for analysis. This feedback loop is used to adjust the MC's state and adapt to internal conditions and external environmental circumstances. An AM could be incorporated into each spacecraft to monitor the on-board systems and self-adjust if necessary. Having an external AM receive feedback from the entire swarm would create a more interconnected swarm, the AM could monitor mission performance and decide to change the mission's high-level global goals and the method used for intra-swarm co-operation.

Swarm research can be categorised in to two topologies, on whether it takes a centralised or decentralised approach to swarm structure [3]. In a centralised swarm some entities are given an organizer role and can control how the swarm behaves. A centralised system can also be referred to as a hierarchical system, this type of system can have many or just one entity in control of a swarm. A swarm that is managed by an external AM controller could be considered a centralised swarm, the disadvantage of this setup is that the swarm is ineffective if the AM is damaged. A hybrid centralised approach could incorporate an AM Controller role into each spacecraft. This would reduce the risk of mission failure, if the AM is lost then another can take its place. To reduce the workload placed on a centralised component, a high level of individual autonomy and autonomicity [4] is needed, intra-swarm communication is also necessary so that co-operation doesn't rely on a central component

In contrast to this, in a decentralised system all entities are equal, there is no controller, the swarm must coordinate its behaviour to complete a task. Distributed communication can involve messages being passed to other swarm members or take the form of prompts left in the environment. The disadvantage of a decentralised swarm is that there is no Manager collecting and analysing global swarm data in situ.

The swarm can be further classified based on the technique used for communication, this can be either direct or indirect [3][5]. Direct communication can take the form of messages sent to other swarm entities or a centralised Ruler/Controller entity. The majority of swarm research focuses on Indirect Communication in the form of environmental triggers. For this research we have chosen to use Direct Communication as it offers more data granularity.

2. RESEARCH BACKGROUND

NASA's Autonomous Nano Technology Swarm (ANTS) is a concept architecture for future spacecraft exploration missions, it takes inspiration from the collective co-operative behaviour of an ant colony [6]. The Prospecting Asteroid Mission (PAM) would involve sending a swarm of 1000 pico-class spacecraft to explore an asteroid belt. The majority of the swarm would be made up of worker spacecraft, of which there would be 10 types each carrying a different scientific instrument such as X-ray, Gamma-ray, magnetometer and infrared sensors [7]. Some spacecraft would not carry a scientific instrument and would instead be given the role of Rulers and Messengers. Rulers are given the responsibility of coordinating the workers, while messengers relay the collated data back to Earth. The swarm would co-operate to complete a high-level goal that has been set by the Rulers. Messengers are given the task of relaying the data back to Earth.

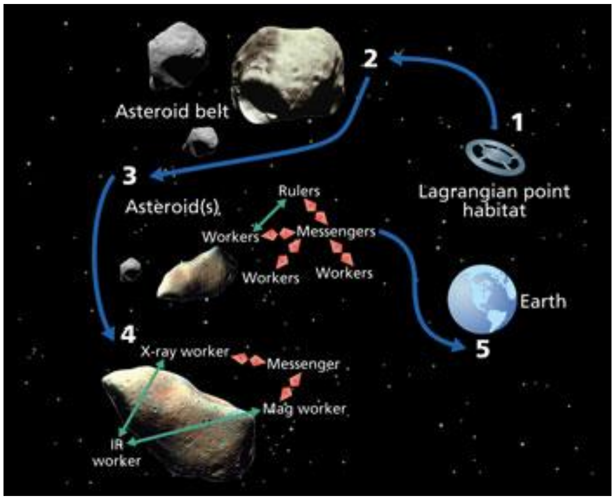


Figure 1. NASA ANTS PAM Mission [1]

The purpose of this research is to investigate which co-operation strategy works best in terms of time to completion and collective work effort. Previous research [8] used the ANTS hierarchical model of Rulers and Workers as inspiration, a simulated 2D environment was created, within which a swarm of Worker robots is controlled by two Ruler Robots. This is an extreme interpretation of the ANTS scenario; the robots were not given much autonomy. If the Ruler decided that a part of the environment required the resources of multiple Workers, it would create a sub-swarm by sending a message to those in its proximity. The robots had to respond unless they had already found an interesting environment feature to explore, in which case they would ignore the request to join the Ruler's sub-swarm.

A second simulation shown in Figure 2 was created, it gave the swarm more autonomy by removing the Ruler. It used a passive Message Board entity to receive and deliver messages between swarm entities, all decisions were made by the swarm as opposed to the Ruler robot. Within the simulation the purpose of the communication was to find a worker that could help another worker analyse an interesting feature that had been discovered in the environment. The swarm were split up into colours, each representing a different instrument, the idea behind this was that more than one instrument may be needed to analyse minerals. This simulation provided every swarm robot with the capability of requesting a sub-swarm unlike the first simulation where this was reserved for the Rulers. However, both swarm models are imperfect in that they are risky and place a lot of responsibility on a few Rulers and a centralized Message Board entity. A model in which the swarm relies on a few Rulers is more vulnerable than a decentralized approach, in the PAM mission less than 20% of the spacecraft would be allocated as Rulers and given the task of coordinating the Workers. With the possibility of 60-70% of the spacecraft being lost during the mission this hierarchical structure could compromise the mission's success [1] [9] [10].

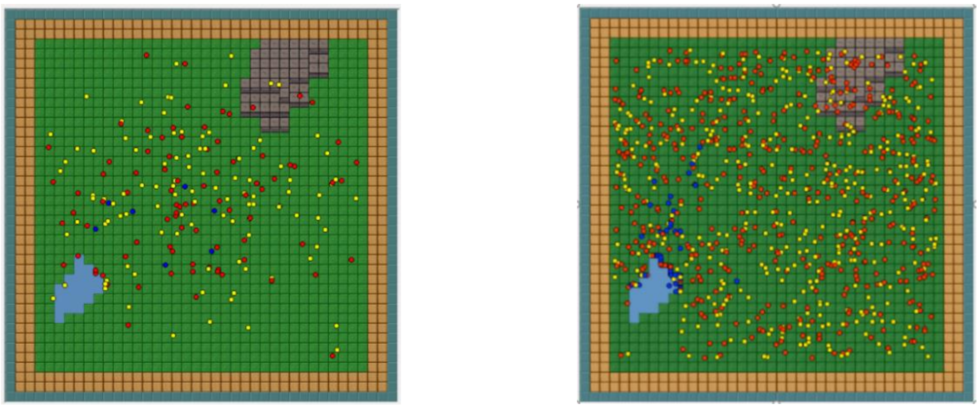


Figure 2. Simulation with Centralised Message Board component. A swarm of red and yellow robots was tasked with finding water and recruiting other robots to help analyse it. Robots change colour to blue when responding.

In contrast to the hierarchical swarm of Rulers and Workers and the centralized Message Board in Figure 2, the simulation in Figure 3 demonstrates message passing between a decentralized homogeneous swarm. The swarm was given the task of searching the map for sources of water. If a robot discovers water, it sends a message to all robots that fall within its signal range. The signal range is determined by the circle surrounding the robot, all neighbours that intersect with the initial robot will receive the message, the message is then passed on to all swarm robots who interlink with the neighbours circles. This forms a message chain throughout most of the swarm, when a robot receives a message it briefly changes its colour to yellow. A robot will respond to a message only if the sender's location is within a certain distance as the robot must have enough battery power to reach the location. When a robot is responding to a message it changes colour to blue and moves toward the location of the robot which sent the message. In Figure 3 there are two water sources, in (a) several robots have flashed yellow to indicate they have received a message. In (b) most of the swarm have moved to the location of the water. This type of swarm behaviour reduces the bottleneck that can occur when using a centralized element like the message board in Figure 2.

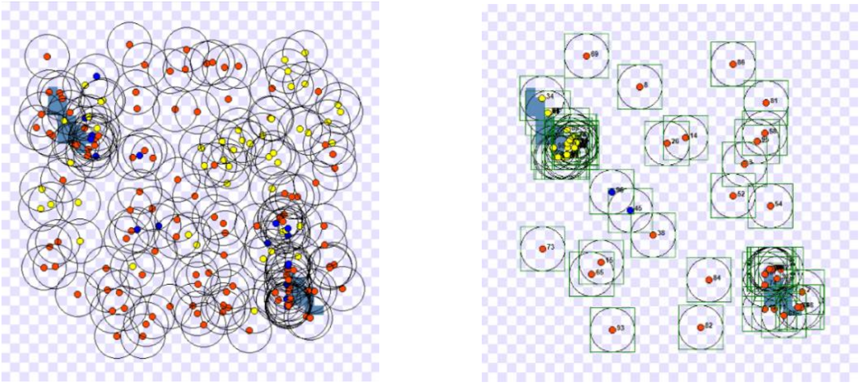


Figure 3. Decentralized swarm with message passing (a)Yellow message passing chain is visible. (b) most of the swarm have flocked to the water sources.

The simulation discussed in this paper explores three co-operation protocols that enable a heterogeneous swarm to collaborate on a task via message passing. The task involves the exploration of a 2D grid. The goal is to find items that require analysis. The items and swarm are split into different colours, in order to analyse an item, the swarm robot's colour must match the item's colour. If the colour doesn't match, then the robot sends a message asking for assistance from other robots that can perform the analysis. The PAM mission would also require this level of swarm co-operation as each spacecraft would only have one specialty [11].

The first protocol is inspired by the Rulers and Workers model in that a robot can request multiple robots to respond to its location by sending a help message to its neighbours, it differs in that all robots can assume the role of a Ruler and request the assistance of other Workers. The message is then passed throughout the swarm between entities which are within range of each other, all robots that have the requested instrument will then move towards the original robot's location. While this protocol provides a high likelihood that an item will be analysed it can also create a mass movement of swarm entities to one location, this can negatively impact the power reserves by increasing the amount of wasted effort and step count.

The second protocol reduces the number of responders that respond to a help message. When a message is received by neighbouring robots, they only forward it on to their neighbours if they do not have the instrument required to analyse the item. This protocol cuts down on the amount of help messages and wasted movement.

The third protocol involves the robot choosing only one robot from those that reply saying they are available to help analyse the item. This protocol has the least amount of wasted resources as only one robot is needed to analyse the item, however wasted effort is still recorded if another robot with the required instrument encounters the item by chance during the time it has taken the responder to reach the item's location.

Future research will look at implementing an Autonomic Manager (AM), this will be a type of Ruler robot that can analyse mission performance and re-configure the swarm in real-time to use a co-operation protocol that delivers best results. If the swarm is low on power, the AM would be able to instruct the swarm to use a protocol that reduces the amount of effort spent on sending communication messages and movement. Protocol 1 has the highest movement cost and communication traffic of the three protocol strategies. The AM would also be able to adjust the make-up of the swarm by analysing the data it receives, if there is an increase in help requests for a certain instrument the AM could send in more craft of that type and withdraw those with instruments that are not in great demand.

Future research will closely prototype the PAM mission architecture by having Rulers coordinating Workers, however instead of having designated Rulers which perform no other function it will incorporate a role-switching Ruler capability into every swarm robot. This would enable the swarm to adapt if an AM robot is damaged, the swarm would collectively choose another robot to assume the role of a Ruler. The role of the AM would be to analyse mission performance by examining the metrics that are sent periodically from the swarm. If necessary it could instruct the swarm to change their co-operation protocol during a mission, it could also change the signal strength setting to increase/decrease the amount of power they use when sending intra-swarm messages. Reducing the amount of power used to send a message will limit the range that the signal can reach, lower signal strength saves power but also lowers the number of robots that will receive the message.

The AM would need to analyse the data to determine whether the environment warrants using a higher signal strength. If a swarm is spread out across a vast area, then it may be necessary to use more power for communication. If battery power is low and the swarm is sparse then using a protocol that requests one or a few responders would be more efficient than wasting resources by having multiple responders move to a one robot's location. The AM could use projection to determine the best course of action, the data being transmitted from the swarm could be fed in to simulations that test each communication protocol with each signal range. This would provide a best-case scenario and allow the AM to continually monitor, analyse, plan and adjust the swarm's high level behaviour [12].

This type of hybrid centralised system with an external AM would be less susceptible to failure than a fully centralised setup. Co-operative swarm communication would still be decentralized in order to avoid a bottleneck of messages being routed via a central controller. All help messages would be sent to neighbours instead of via the AM. The level of swarm co-operation could be adjusted by the AM, if the analysis of the data indicates that a lower signal strength would suffice then a message could be sent to the swarm to request that it change its signal range. The co-operation protocol could also be switched, if the swarm is running low on battery life then the AM could instruct the swarm to use a protocol that is less wasteful.

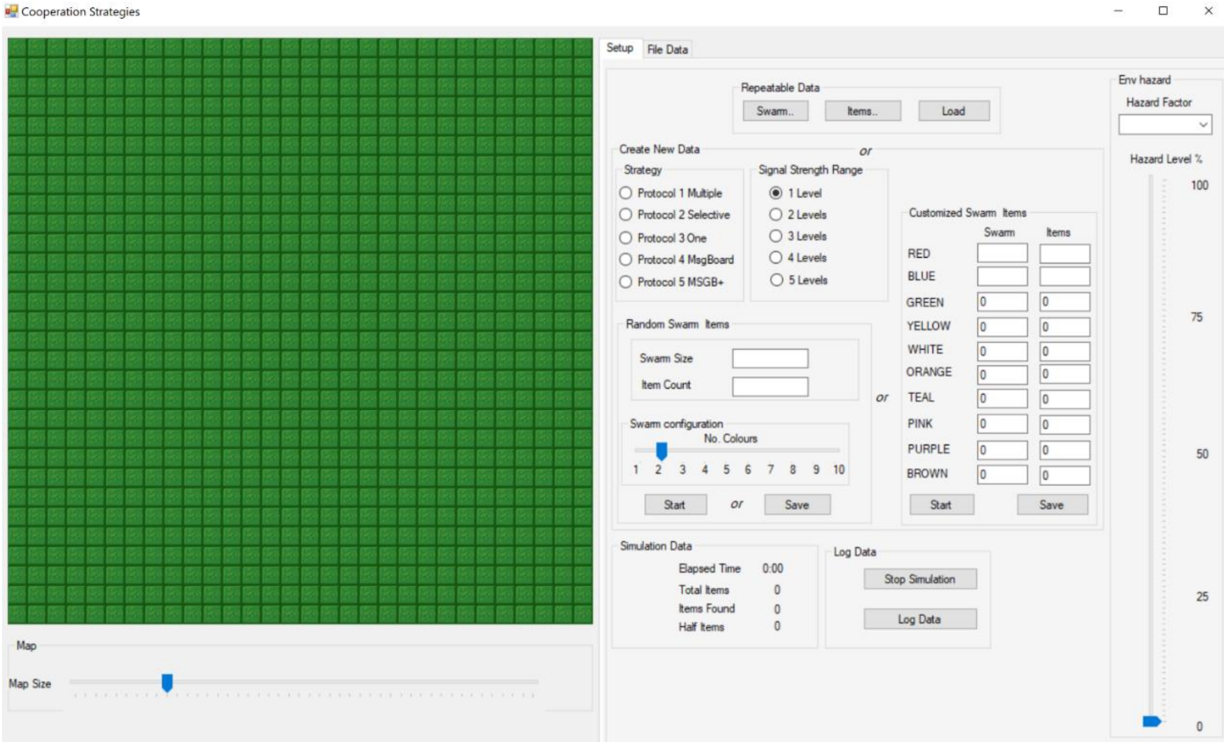
3. SIMULATION DESIGN

The simulator's GUI is shown in Figure.4, it was written in C# and consists of a grid of 900 cells and a form to input the swarm size and choose which communication protocol to use. The grid is made up of a 2D array of 900 cells, each cell's row and column is used by the swarm to traverse the grid. A map cell holds information about position, the items that it contains, and the type of cell that it is. For this simulation all tiles are homogeneous and traversable, previous experiments used untraversable tiles to simulate obstacles such as water and mountains.

The swarm can be configured for 10 instruments, their specialty is depicted by a colour, the items are also colour coded to represent the instrument needed to analyse it. The swarm are given the task of exploring the map, they can analyse items that match their colour but need to request assistance when they discover an item that doesn't match their colour.

3.1 Form Design

- Strategy Type – 3 co-operation protocols {One Responder, few Responders & Multiple Responders}
- Signal Strength – how big the signal range is, lowest is 1 level around the robot, up to 5 levels
- Swarm Size – the number of swarm robots.
- Item Count – the number of items scattered within the environment.
- Swarm Configuration – the number of colours/instruments that the random swarm is divided into.
- Customized Swarm & Items – allows for precise input of swarm and items colours/instruments.
- Hazard Level – Simulated failure, the level% indicates how hazardous the environment is.
- Hazard Factor dropdown box allows the user to choose how often the robots encounter a hazard. It is possible to simulate a hazardous environment by increasing the likelihood of dying (increasing the level toward 100%) and decreasing the Hazard Factor so that the time window between encountering a hazard is shorter. The robots are then more likely to die and often.



○ Protocol 1 Multiple
○ Protocol 2 Selective
○ Protocol 3 One
○ Protocol 4 MsgBoard
○ Protocol 5 MSGB+

● 1 Level
○ 2 Levels
○ 3 Levels
○ 4 Levels
○ 5 Levels

Swarm Size
Item Count

	Swarm	Items
RED		
BLUE		
GREEN	0	0
YELLOW	0	0
WHITE	0	0
ORANGE	0	0
TEAL	0	0
PINK	0	0
PURPLE	0	0
BROWN	0	0

Elapsed Time 0:00
Total Items 0
Items Found 0
Half Items 0

Figure 4. Simulation user interface.

The user can run the simulation in either random or customized mode, the latter allows for the composition of the swarm to be chosen. The exact numbers and colours of robots and items can be inputted. If the user creates an item and does not create at least one robot that matches the item's colour, then the item will never be analysed and the simulation will run until the robot terminates due to a hazard. However, for these experiments the hazard has been set to zero, future work will look at analysing how hazardous environments can affect swarm performance A random swarm is configured by moving the '*No. Colours'* slider and inputting the overall swarm size, the configuration is then created by dividing *'Swarm Size'* by *'No. Colours'*.

The swarm are represented by squares and the items by circles, up to 10 different colours can be chosen for both. When an item has been analysed its colour changes to grey, as shown in Figure 5 (a) Robots move one cell at a time, they check that cell for items, if an item is discovered it is either analysed or the robot sends a help message to neighbours. When a robot responds to a help request it displays a yellow beacon around itself, in Figure 5 (b) Robot 54 is surrounded by a yellow circle indicating that it is responding to a help message.

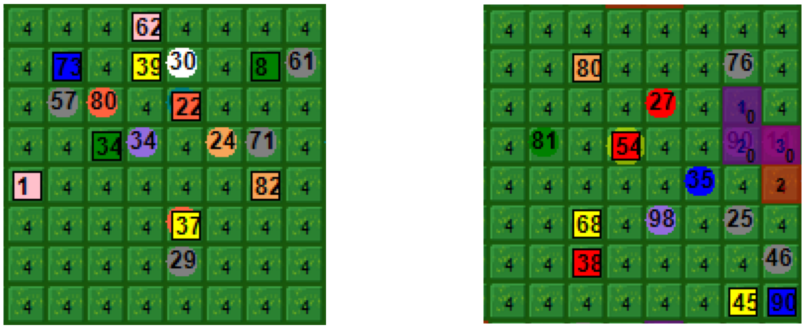


Figure 5 – (a) Robots and Items. (b) Robot 54 responding to a help message.

3.2 Protocols

Most artificial swarm systems try to mimic the collective behaviour of natural systems such as flocking, shoaling and ant pheromone trails. We have implemented a more direct approach to collaboration, messages are sent between swarm entities, each entity is capable of rerouting others to their location. We have created 3 collaboration protocols, they each use message passing to relay information about items that require analysis, they differ in how many robots will respond and move towards the item's location. Only one co-operation protocol can be chosen during each simulation run, it cannot be changed on the fly, future work will look at incorporating an Autonomic Manager that can change the co-operation type as the simulation is running.

3.2.1 Protocol 1 Multiple Responders

This protocol has the greatest reach, help messages are passed to all neighbours within range, they in turn pass it to their neighbours. All robots that match the criteria outlined in the message will move towards the item's location unless they are currently responding to another help request. While this ensures that an item will be analysed it also incurs a large number of wasted steps aka non-productive effort. This occurs when an item is analysed by another robot before responders can reach the item's location.

3.2.2 Protocol 2 Selective Responders

To limit the number of responders, the message is passed to the robot's neighbours but is not always forwarded on. The neighbours only pass it on to their neighbours if they do not have the correct specialty for the item that has been discovered. This reduces the amount of power spent on intra-swarm communication.

3.2.3 Protocol 3 One Responder

This protocol is similar to Protocol 2 in that there is more decision making about whether a message will be flooded throughout the swarm. The messages are only sent to the immediate neighbours and are not forwarded to the rest of the swarm. This is riskier than Protocol 2 as there is a chance that none of the neighbours will be able to help. Those that receive the message and are able to analyse the item send an affirmative response to the sender. The robot then chooses a responder from those that have replied, if after a period of time the robots who responded do not receive a response, they cease waiting and resume their exploration task. The One Responder strategy has the lowest number of responding and wasted steps. It is the only strategy that does not perform any sort of message forwarding between neighbours except for the initial message pass from the Robot to surrounding neighbours. This type of strategy would be useful if the swarm is low on power or the swarm is large and there are numerous robots of all specialties close by. A small area with an abundance of items lowers the need for communication, when the *'found by change'* and *'non-productive effort'* data is high there is less need for collaboration.

If the Productive Effort (PE) is high then there is more of a need for collaboration, PE will be higher if the swarm is small or spread out, low PE and high NPE is a result of too much unnecessary collaboration between a swarm that is exploring a smaller area. High PE and low NPE indicates that robots answering messages are succeeding in getting to and analysing the item before another robot happens upon it by chance, The Multiple Responders and Selective Responders strategies have the most amount of Total Responding Steps as a result of the increased message reach throughout the swarm. When communicating using the Multiple Responders strategy, the amount of Responding Steps and Total Non-productive effort (NPE) is slightly greater than that of the Selective Responders, and both are much greater than the NPE of the One Responder strategy.

3.3 Signal Strength

This is the range that a message can reach when sent from one robot to the rest of the swarm, for these experiments we compare the metrics outputted when running the simulation with signal strength 1, 3 and 5 levels. A level refers to the border surrounding a robot, all three levels are depicted graphically in Figure 6.

The distance each signal strength can cover is shown in Figure 6, the blue area surrounding the red robot depicts the range of the signal strength. When the signal is set to one level it can only reach robots that are positioned in the 8 neighbouring cells. A signal strength of 3 results in a range of 48 surrounding cells. 120 cells are reached when the strength is set to 5, in this case the blue robot would be in range to receive a message from the red robot. If L equal the signal strength level then the number of neighbouring cells will be $(L+L+1)^2 - 1$.

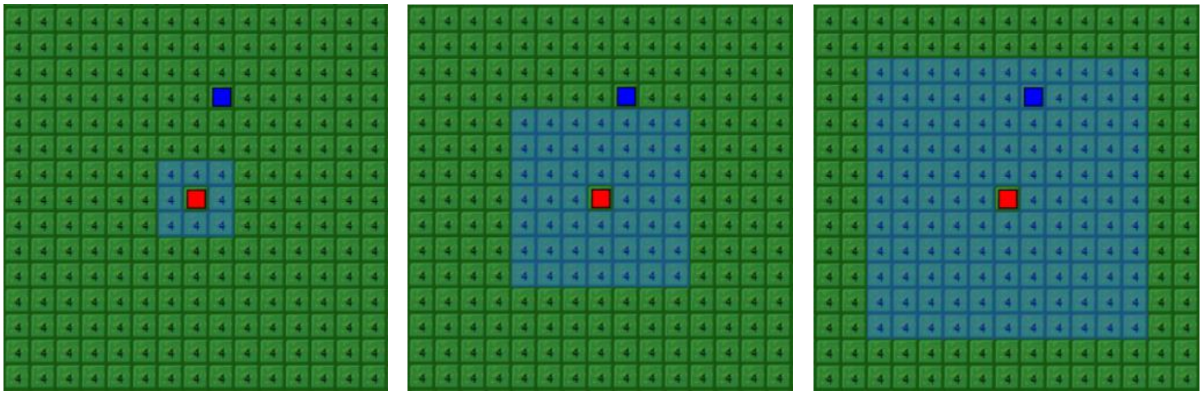


Figure 6. The blue area surrounding the red robot depicts the range of the signal strength. (a) Signal strength is set to 1 level, this covers the cells immediately surrounding the red robot. (b) When the signal is set to 3 the range is 48 cells and 3 levels around the robot. (c) A Signal of 5 has a reach of 120 cells, in this scenario the blue robot would be in range to receive a help message from the red robot.

3.4 Movement

To perform their exploration task, each swarm robot generates a random target location, once they reach the target they create a new target cell. When moving around the grid, robots calculate the shortest path from their cell to the target cell. This is also the case when responding to a help message, the item's location is used as the target. The swarm uses Dijkstra's algorithm to find the shortest path to the target cell. The cells of the grid are connected by edges with a weight of 1, the path that requires the least number of moves is chosen. In Figure 7 (a) the blue robots have generated a random target to move to, the yellow paths show the projected shortest path to the destination cells. As the robot moves along this path it changes each visited cell to blue. In Figure 7 (b) one of the blue robots has reached the end of the shortest path projection, at this point it will now choose a new target cell. As the robot moves across the grid it scans each visited cell for items, it keeps a record of all items that it discovers. It also keeps track of how many items it has analysed and their identification number.

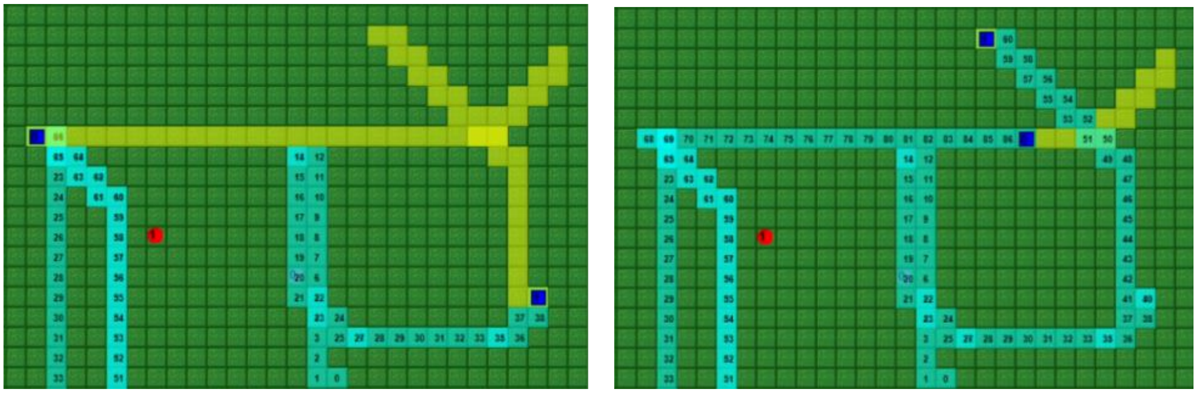


Figure 7. Traversal using the shortest path (a) The projected shortest path is indicated by the yellow squares. (b) As the robot moves along the path it leaves a blue trail and a number to indicate how many steps it has taken.

3.5 Wasted Non-productive Effort vs Productive Effort

The number of overall steps taken by the swarm when it responded to a help request is recorded as either *Wasted Non-productive Effort (NPE)* or *Productive Effort (PE).* If an item is unanalysed when a responding robot reaches its location then the steps are added to the PE metric. However, if another robot had already reached and analysed the item then the steps are categorised as wasted effort.

Figure 8 shows how both are depicted within the simulation, the red trail represents PE, as the robot moves it creates a trail behind it. The red trail cells are also marked with a number to denote how many steps have been taken when responding. If the item has already been analysed by another robot then the colour of the trail is changed to purple and the trail cells count is added to the NPE. Each robot keeps NPE and PE arrays which record the cells visited when responding to help messages.

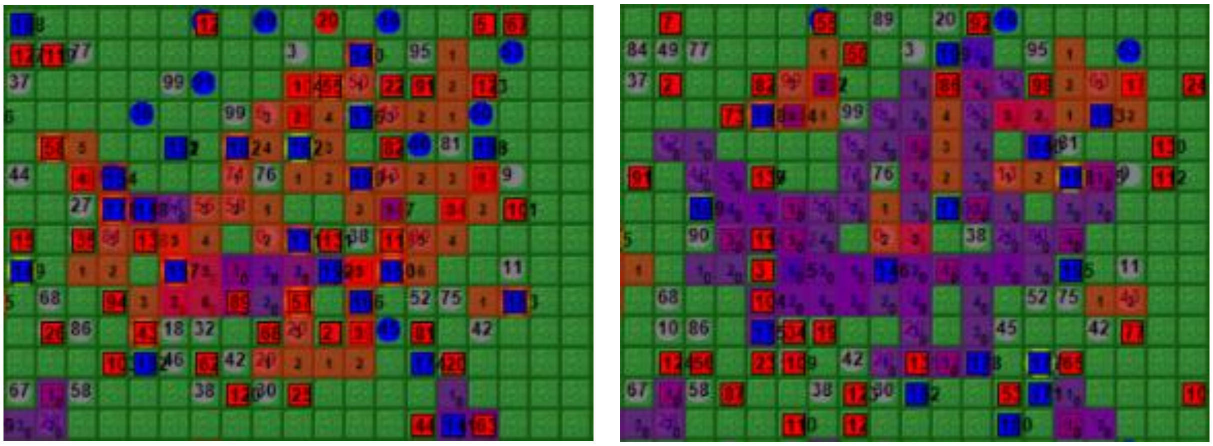


Figure 8. (a) The red cells represent productive effort. The red trail is created when the robot responds to a help message. (b) The purple cells indicate wasted effort. The red trail is changed to purple if the item has already been found and analysed by another robot in the time taken to reach its location.

3.6 Autonomic Manager (AM)

An AM would need to analyse the data to determine whether the environment is rich in items/minerals, if items are in abundance then Protocol 3 is the logical choice as only 1 robot is diverted off course. If the swarm is exploring an environment that is large and lacking in minerals it will take more time to find all items, the more distance it must cover the less likely it will find items that match its specialty. The decision an AM would need to make is whether it is worth using a protocol that invokes mass movement to one item and reduces overall swarm power or to risk no one responding to the message in an effort to save battery power. Giving item types a priority would also help to decide which strategy the swarm should be using. If most items being discovered are low-priority, then conserving battery power should take precedence.

3.7 Help Message Flow Diagram

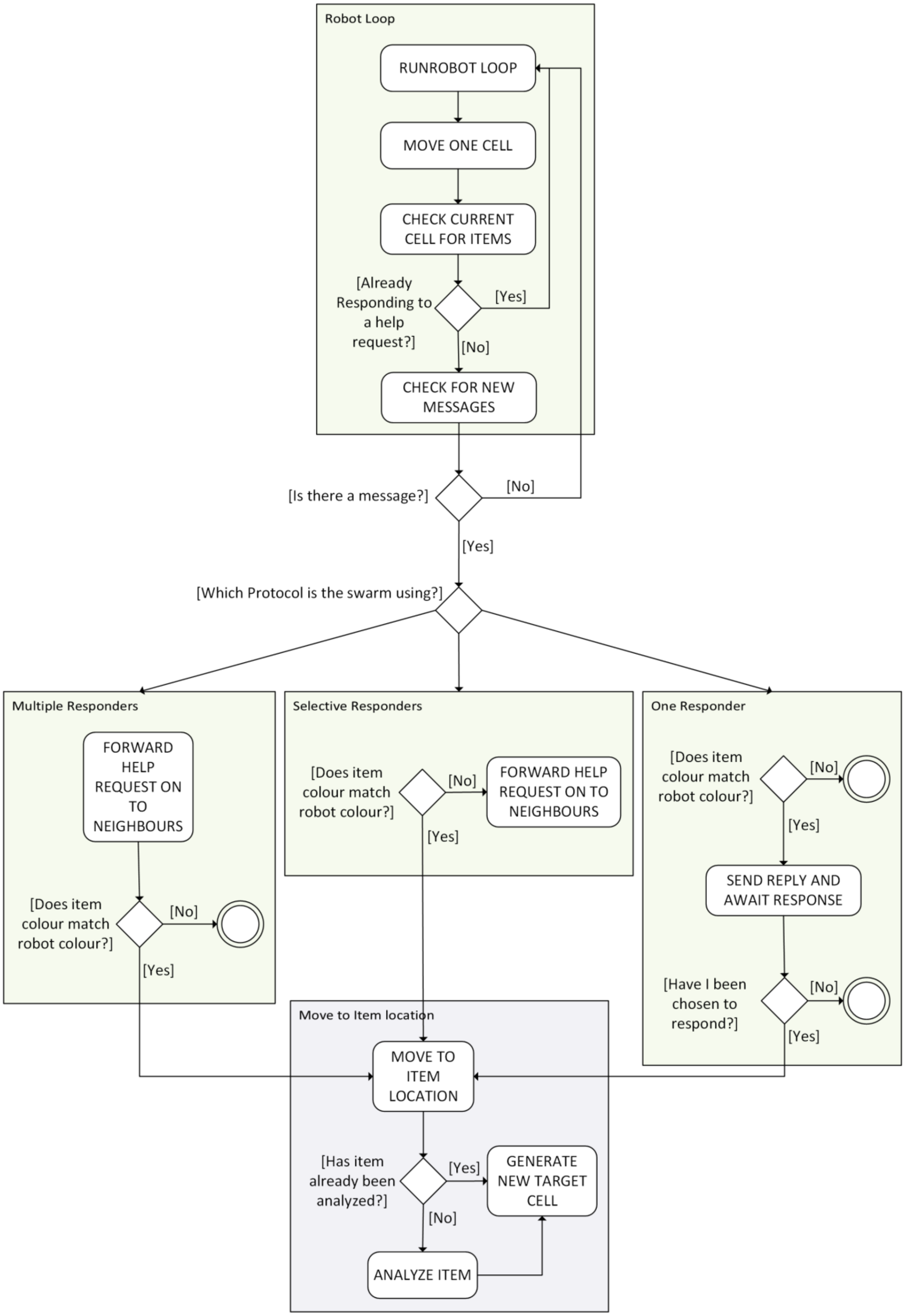

Robot Loop
RUNROBOT LOOP
MOVE ONE CELL
CHECK CURRENT CELL FOR ITEMS
[Already Responding to a help request?]
[Yes]
[No]
CHECK FOR NEW MESSAGES
[Is there a message?]
[No]
[Yes]
[Which Protocol is the swarm using?]
Multiple Responders
FORWARD HELP REQUEST ON TO NEIGHBOURS
[Does item colour match robot colour?]
[No]
[Yes]
Selective Responders
[Does item colour match robot colour?]
[No]
FORWARD HELP REQUEST ON TO NEIGHBOURS
[Yes]
One Responder
[Does item colour match robot colour?]
[No]
[Yes]
SEND REPLY AND AWAIT RESPONSE
[Have I been chosen to respond?]
[No]
[Yes]
Move to Item location
MOVE TO ITEM LOCATION
[Has item already been analyzed?]
[Yes]
GENERATE NEW TARGET CELL
[No]
ANALYZE ITEM

Figure 9. Robot decision making flow diagram.

4. SIMULATION EXPERIMENTS

Three tests were used to compare how the metrics vary when the colour of a binary swarm is evenly split and unevenly split, i.e. when there are more red robots than blue robots. The item count was set to 200, with 100 of each colour of item, 100 red and 100 blue, this was the same for all three tests. The metrics that are recorded include *'number of help requests sent', 'number of help requests answered', 'items found as a result of a help request', 'Total Responding Steps', total non-productive effort', 'productive effort', 'total steps'*.

4.1 Test Scenarios

- Test 1 Split – 50% red 50% blue (100 red, 100 blue)
- Test 2 Split – 70% red 30% blue (140 red, 60 blue)
- Test 3 Split – 90% red 10% blue (180 red, 20 blue)

All three protocols were run for each test, for each protocol, 3 signal strengths (1, 3 and 5) were tested. This resulted in 27 different test scenarios, each were run 10 times and the results were then averaged. Comparisons were then made between how each Protocol performs across all 3 tests, how the metrics differ when the signal strength is changed and how the protocols compare to each other for each test.

Figure 10 shows the legend that is used to differentiate between the 27 test scenarios. The name of each test is derived from the co-operation Protocol number (1: Multiple Responders, 2: Few responders, 3: One Responder), followed by signal strength and then the Test colour split. Test 1 is described as 50-50, Test 2 is 70-30 and Test 3 is 90-10, this represents the percentage of each swarm colour. Each of the 3 tests is further defined graphically by either a circle, diamond or triangle, the colour scheme is arbitrary.

4.2 Legend

- Circle: Test 1 50% red (100 robots) and 50% blue robots (100 robots)
- Diamond: Test 2 70% red robots (140 robots), 30% blue robots (60 robots)
- Triangle: Test 3 90% red (180 robots), 10% blue (20 robots)

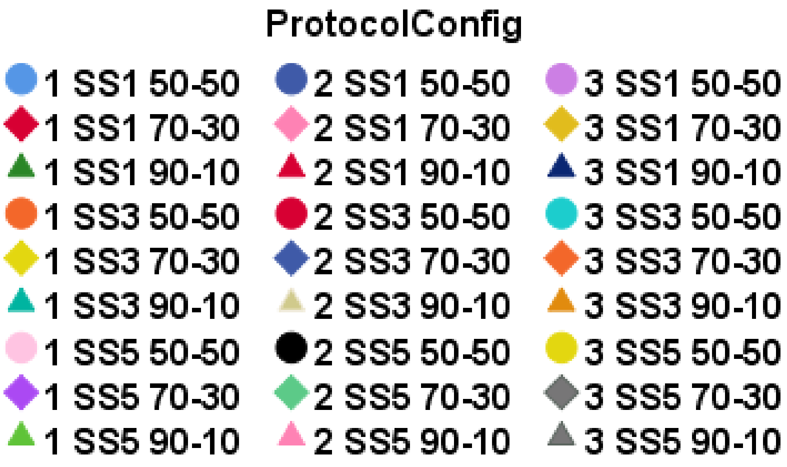


Figure 10. Legend for Tests 1, 2, 3.

5. SIMULATION ANALYSIS

All results were averaged and imported into SPSS for analysis. The following section uses some of the metrics that were outputted from the simulation and compares the performance of the protocols. The metrics discussed are *'Total Help Requests Sent'*, *'Number of Help Requests Answered'*, *Total Steps and Simulation Time'*, and *Non-productive Effort'.*

5.1 Total Help Requests Sent

For tests 1, 2 and 3 the number of help requests sent is increased when the signal strength is lower. This is the case for all three protocols and the three colour split tests (50% red 50% blue, 70% red 30% blue, 90% red 10% blue). When the signal range is set to five the number of help messages sent is less than if it was set to one. As the signal has more reach, more help messages are reaching robots that can respond to an item. This reduces the number of times that multiple robots encounter the same item and send a duplicate help message for the same item. In Figure 11 the *'TotalHelpRequests'* for Test 3 Protocol 1 with a split of 90-10 declines as the signal strength increases.

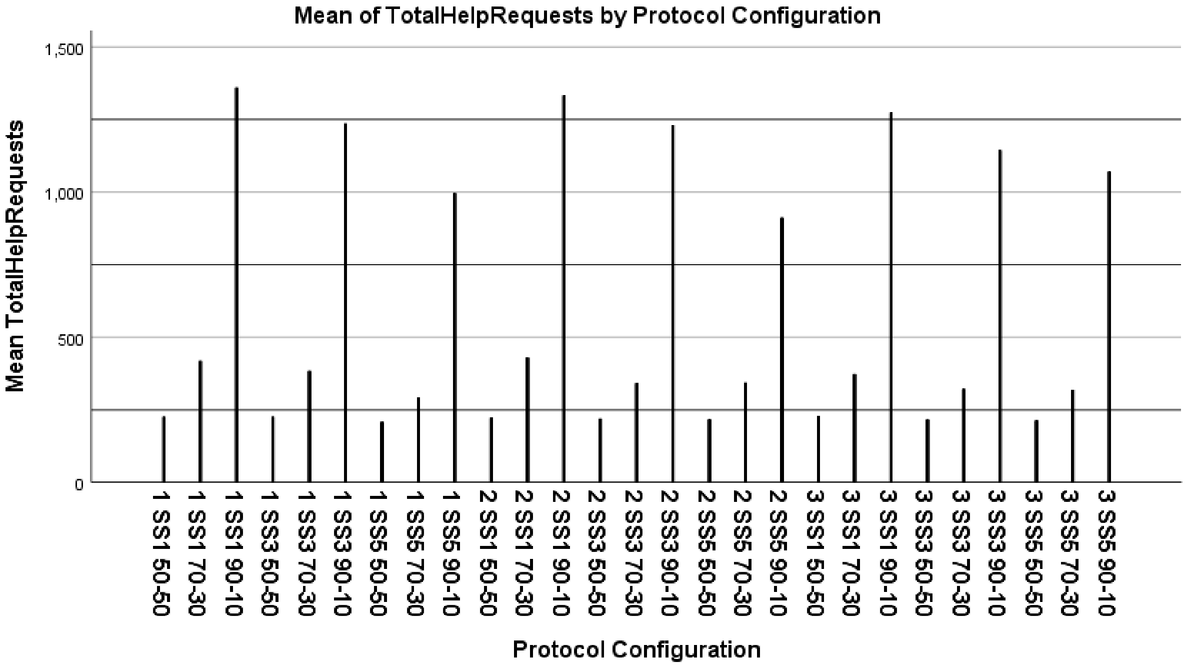


Figure 11. Mean Total Help Requests that were sent by each Protocol for each test split.

5.2 Number of Help Requests Answered

An equal distribution of colours/instruments results in a much lower number of help requests sent than in Tests 2 and 3. The number of help requests answered varies depending on the swarm colour compostion, with tests that use a higher signal strength resulting in the lowest *'TotalHelpRequests'* and highest *'TotalHelpRequestsAnswered'*, this is due to an increased reach which increases the chance of the help request being answered. The greater the signal reach and increased number of responses, the less requests are made by multiple robots for the same item. If signal strength is high, the amount of overall power wasted on sending requests is lower.

An Autonomic Manager would need to consider whether sending many requests more frequently across a shorter distance consumes more power than sending fewer requests over a greater distance. Protocol 1 has a consistently high number of*'TotalHelpRequestsAnswered'* across all 3 tests, the greater the signal reach the more robots will be required to respond and move toward the item's location. If battery life is limited, then using a protocol which doesn't divert large numbers of swarm robots is more prudent. Protocol 3 answers the fewest requests, especially when the signal strength is at its lowest. If minerals are in abundance and the battery life of the swarm is finite, then this protocol is the best choice as it will save power while also performing a basic level of co-operation.

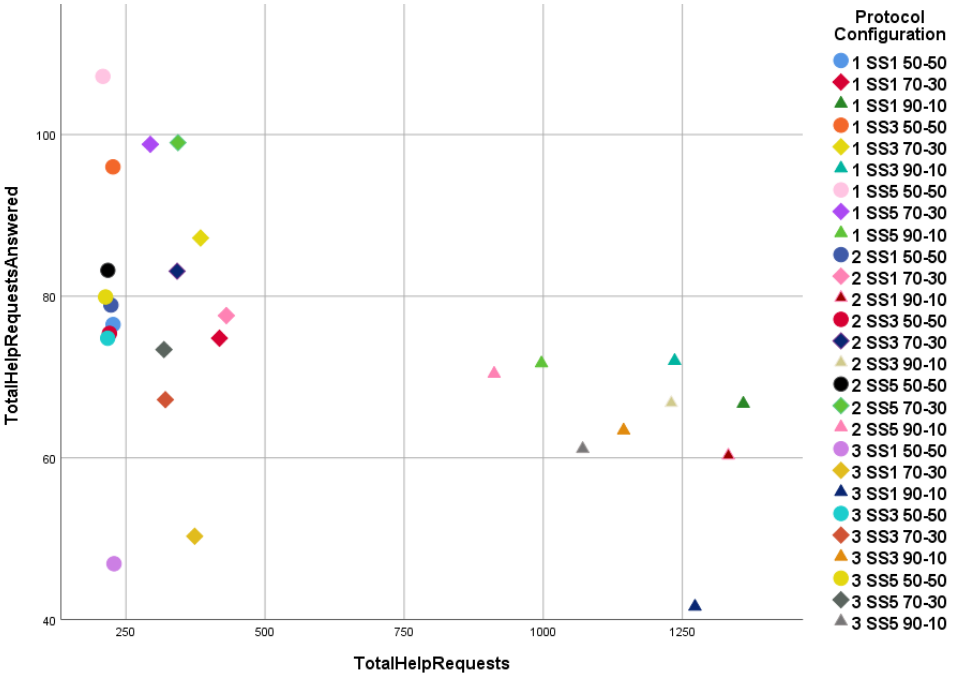


Figure 12. The number of Help Requests that were answered compared to the Total Help Requests sent. The tests are shown as follows; Circle depicts Test 1, an equal swarm; Diamond represents Test 2, an unequal swarm; Triangle signifies Test 3, an extreme swarm configuration with 90% of one colour.

5.2.1 Test 1 – Equal Configuration (50%-50%)

Protocol 1 with a signal strength of 5 is depicted as a pink circle in Figure 13, it has the lowest number of help messages sent and the highest response rate. As more help messages are answered this reduces the number of duplicate messages being created when an item is rediscovered by other swarm entities. The higher signal strength increases the reach of the message and results in an increased chance of a robot replying to the message. By contrast, Protocol 3 with a signal of 1 level offers the lowest amount of co-operation, more help messages are created but it has the least number of replies.

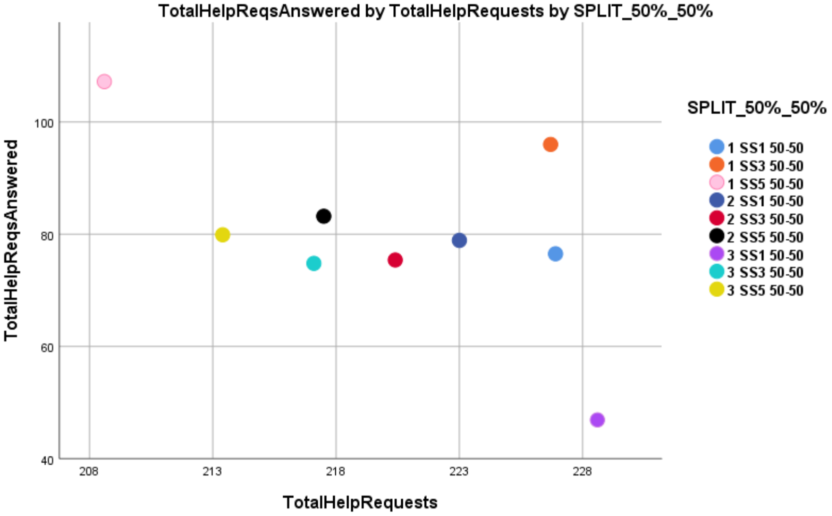


Figure 13. Test 1, number of Total Help Requests Answered compared to Total Help Request sent.

5.2.2 Test 2 – Unequal Configuration (70%-30%)

The maximum number of help requests sent is nearly double those sent when the swarm is split 50-50. This is to be expected as the items are still split evenly which increases the workload for the robots in the 30% category. Protocol 1 with a range of five is again the most efficient in terms of the ratio of request answered to requests sent, it is depicted as a purple diamond in Figure 14.

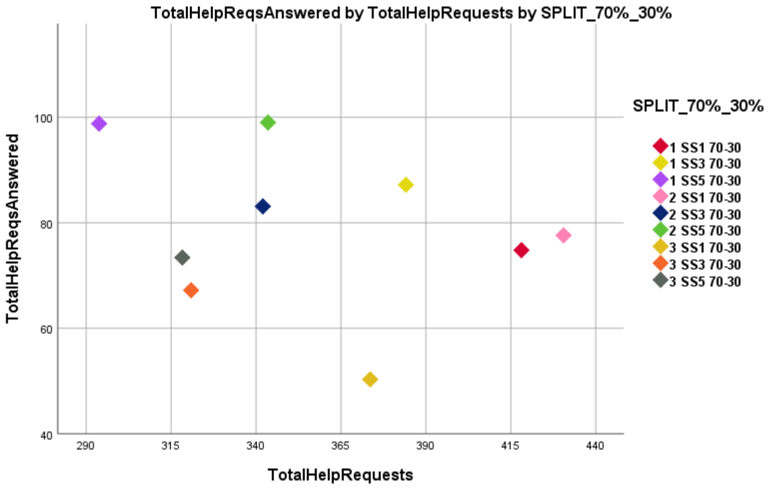


Figure 14. Test 2, number of Total Help Requests Answered compared to Total Help Request sent.

5.2.3 Test 3 – Extreme Configuration (90%-10%)

When the swarm is configured with an extreme mix of 90% one type of robot and only 10% of the other colour there are significantly more help requests sent. Figure 15 shows that Protocol 1 (Multiple Responders) has the highest answered rate, this occurs when using a signal strength of 3 or 5. This is closely followed by Protocol 2 when using the higher signal ranges. The least number of requests answered occurs when using Protocol 3 (One Responder Protocol) in conjunction with the lowest signal strength. This is due to the reach being reduced as the message is only sent to the robot's neighbours, it is not forwarded to the rest of the swarm. If there isn't a robot within the vicinity that is able to respond to the request, then the message goes unanswered. With the swarm being configured to have 90% of one type there is less likelihood of having a robot from the other type as a neighbour. In Protocol 3 the message is not passed on, the co-operation effectiveness when using a low signal range is much less than when using a higher range.

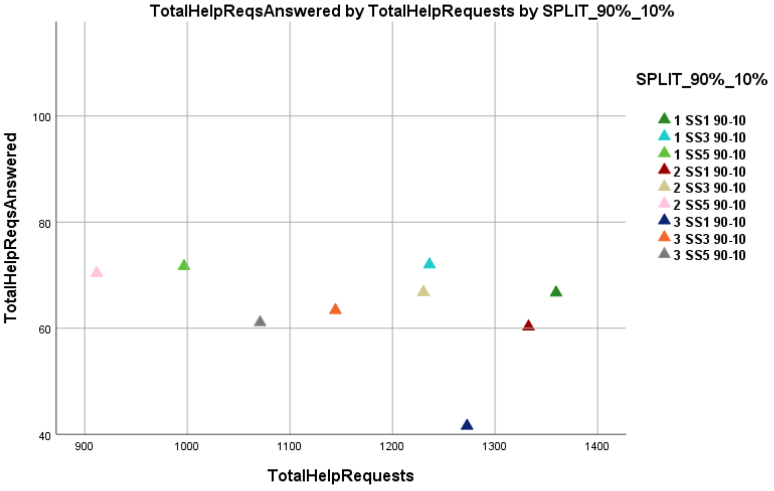


Figure 15. Test 3, the number of Total Help Requests Answered compared to Total Help Request sent.

5.3 Total Steps and Simulation Time

For an evenly split swarm of 50%-50% the results don't vary much between protocols or signal strength, Protocol 3 is the fastest when the signal is set to 5. This is most likely due to the fact that multiple robots are not being diverted to one point on the grid, consequently reducing their ability to find new items and adding more time to find all of the other items not listed in the help message. For Test 1 with a 50%-50% split, Protocol 3 with a signal of 5 finishes in the fastest time and lowest step count.

For a swarm that is split 90%-10% the most inefficient configuration is Protocol 3 when combined with a signal range of 1 level. This protocol allows only 1 responder for each help message, the message is never passed on by neighbours, therefore Protocol 3 has the lowest chance of a help message receiving a response. When the signal is upped to 5 this protocol becomes the more efficient, finishing in half the time. It is shown as a grey triangle in Figure 16, the rest of the configurations for Test 3 take longer and use more steps. This is something to consider when a swarm has a finite number of robots with a certain instrument, more signal reach is required and having one responder is more efficient due to a limited pool of robot resources.

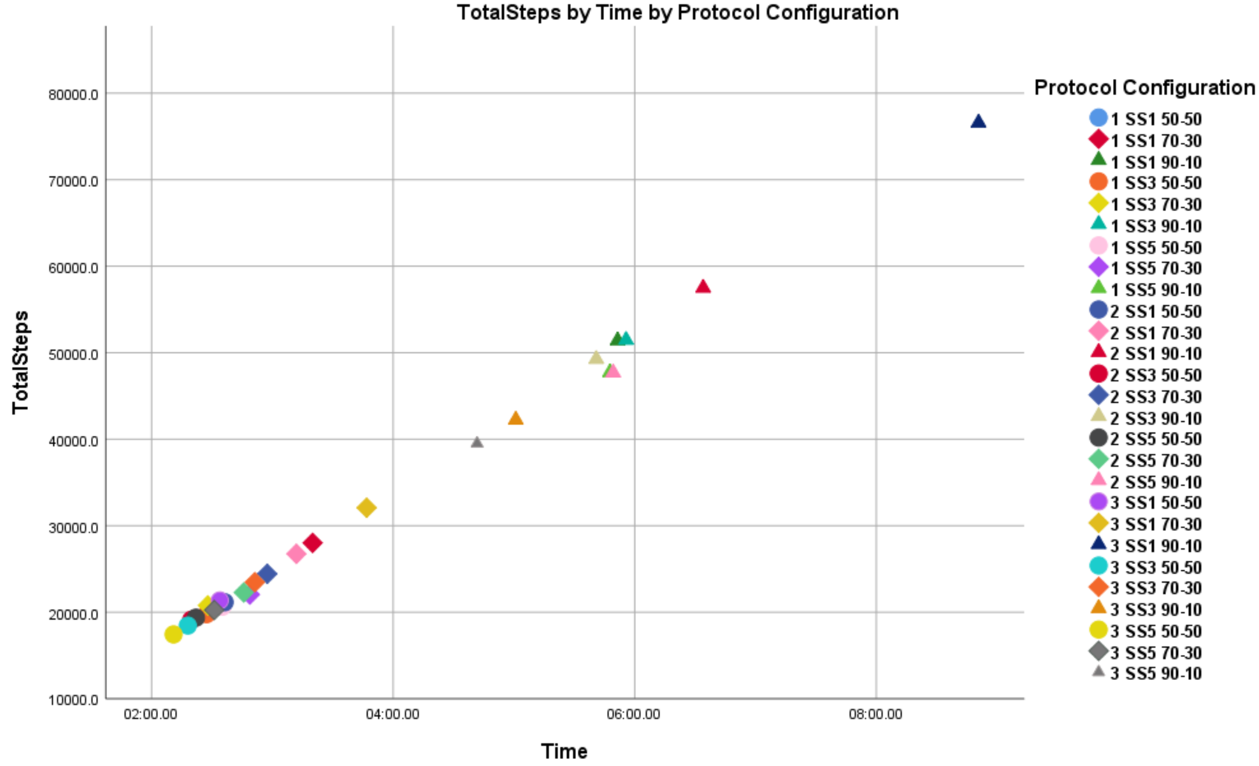


Figure 16. Scatterplot showing how step count increases when the simulation time is longer. Test 3 has the longest simulation running time and therefore the highest step count.

5.3.1 Test 1 – Equal Configuration (50%-50%)

The mean simulation running time is less than 3 minutes when the swarm is evenly split between red and blue items. The number of steps taken is the fewest of all the tests at around 20000 steps mark. The time difference isn't that great between the protocols, however more co-operation tends to negatively impact the run time. If simulation running time and battery life are not a high priority, then more time and effort can be spent on co-operation and responding to help messages. A highly co-operative swarm would ensure more items are found and analysed. An AM would need to track overall swarm battery life and assess whether the swarm can run in a highly co-operative mode or whether having just one responder is best. It would also need to monitor how far the swarm is spread out and devise the optimum signal strength to ensure at least one robot responds to each help message.

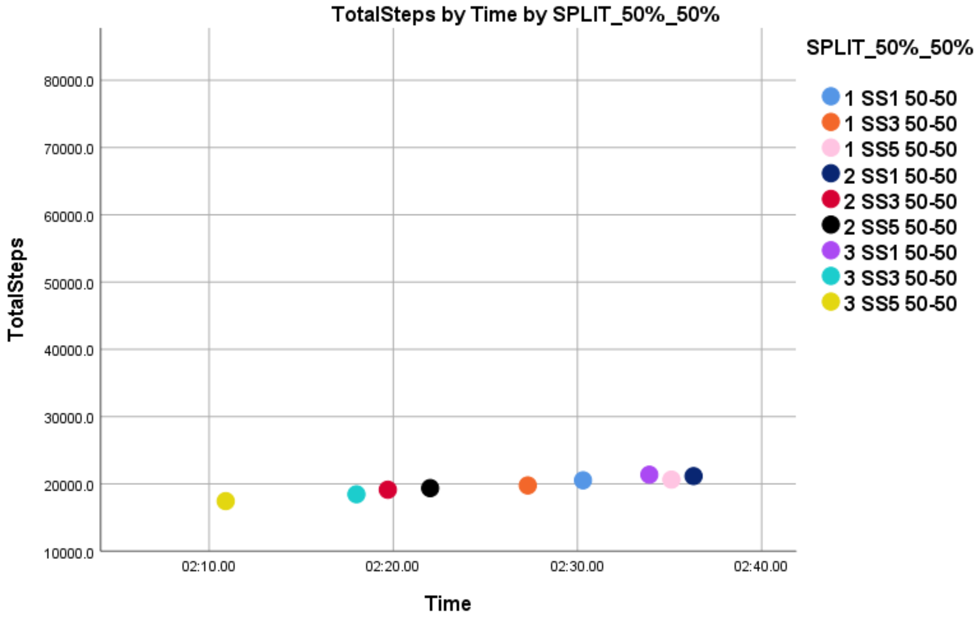


Figure 17. Test 1's Total Steps and Time metrics for all protocol configurations.

5.3.2 Test 2 – Unequal Configuration (70%-30%)

The mean maximum simulation time is over a minute greater than when the swarm is evenly split. This increases the overall step count due to the additional time required to search the grid. When all 3 protocols use the lowest signal strength the simulation takes longer to complete. In Figure 18 the red, pink and gold diamonds have the longest completion time, they represent Signal Strength 1 for the 3 Protocols. This shows that very little co-operation increases time and increases step count which negatively impacts battery life.

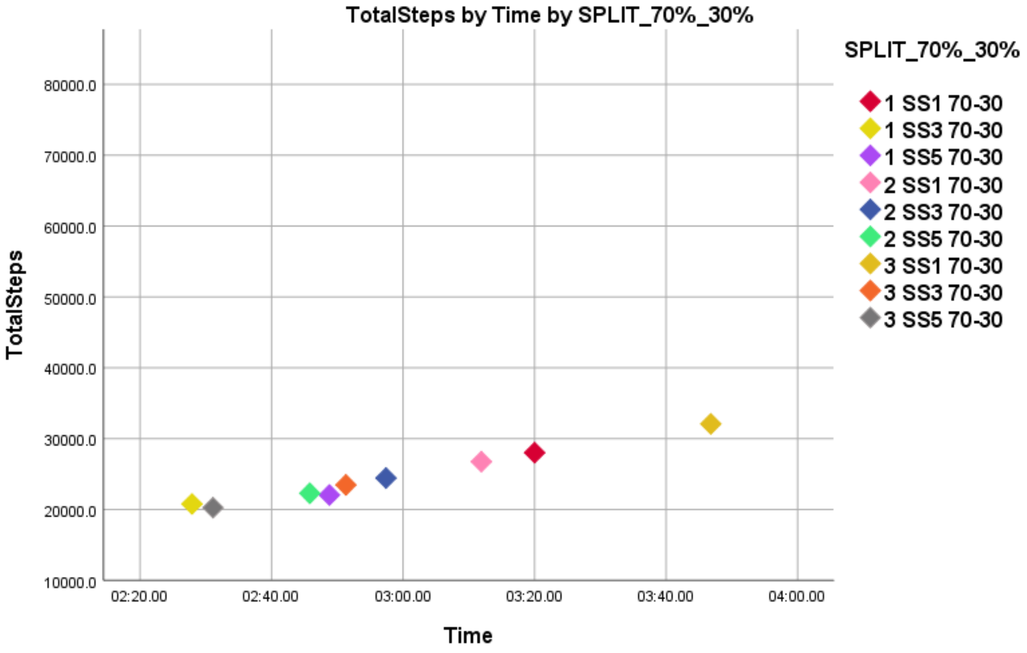


Figure 18. Test 2's Total Steps and Time metrics for all protocol configurations.

5.3.3 Test 3 – Extreme Configuration (90%-10%)

The greatest variety in performance for Test 3 can be seen in Protocol 3, using the lowest signal strength results in an average simulation time of 8:50 minutes, whereas the average time for the highest signal is 4:40 minutes. This protocol only allows robots to send messages to neighbours that fall within the signal range, the message cannot be passed on by those that receive it. As only 1 responder robot is chosen, the wasted effort is reduced but if a low signal is used then the number of duplicate help requests is greater as the message is less likely to be responded to. The overall total step count increases as the time increases. When the signal range is at 5, Protocol 3 is the most efficient in terms of time and step count.

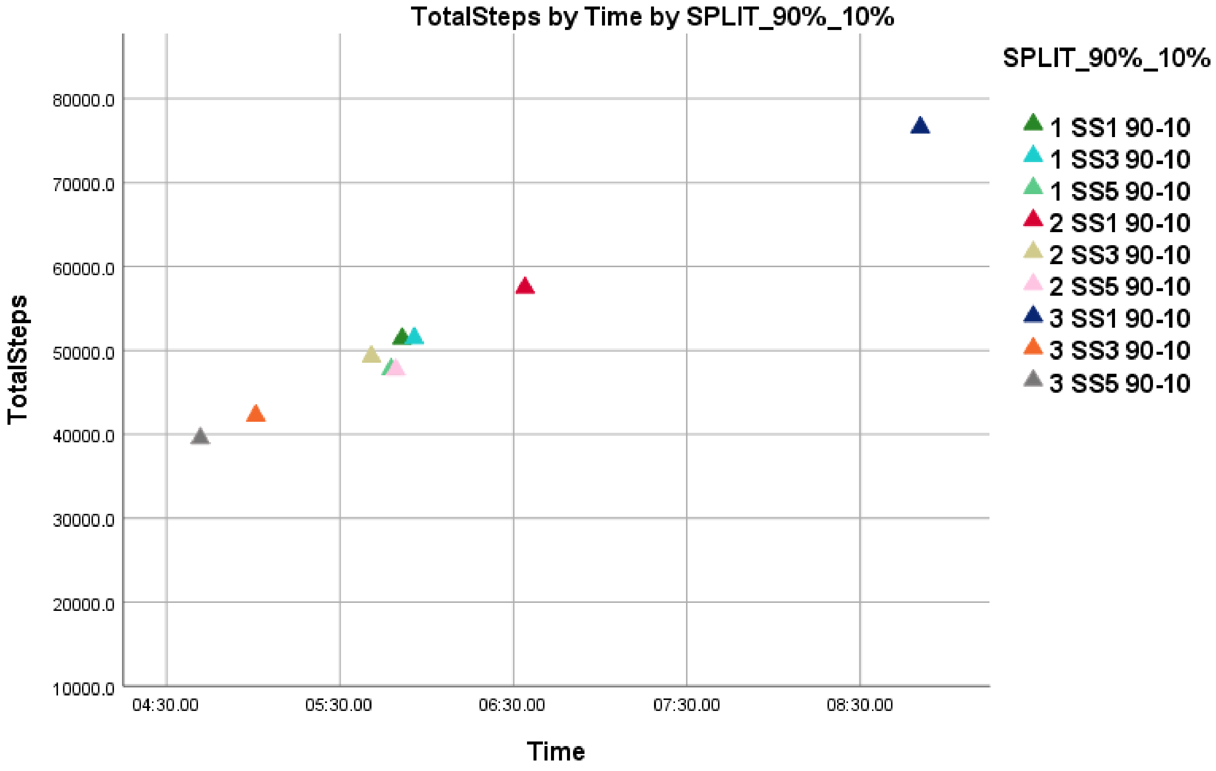


Figure 19. Test 3's Total Steps and Time metrics for all protocol configurations.

5.4 Non-productive Effort (NPE)

When the level of co-operation is increased, the amount of wasted non-productive effort is also increased. Using a higher signal strength causes an increased signal reach and greater *'TotalRespondingSteps'* and *'TotalNPE'*. Test 3 results in the least amount of NPE due to there being less co-operation overall and therefore fewer *'TotalRespondingSteps'*. When the swarm is extremely uneven there is less chance of the message reaching the 10%, this reduces the wasted effort but negatively impacts simulation time. Across all 27 configurations there is a negative correlation between the number of *'TotalRespondingSteps'* and the total simulation time. A lower signal range causes a reduced amount of responses, when 'TotalRespondingSteps' is low then simulation *'Time'* is high.

Having only one responder per message cuts down on the amount of NPE that occurs when multiple robots respond to the same help message. The fewer responses there are the less non-productive effort, this reduces the unnecessary movement cost that occurs when an item is analysed before a robot can reach it. Reduction in movement cost saves battery power, if the minerals are in abundance and the environment is low-risk then sending one responder is acceptable. If the environment is hazardous then the AM would need to factor in to its projections the likelihood of the responder reaching its destination. If the data indicates a hazardous environment then sending more responders may be more practical, despite the fact that this will drain the battery life of the swarm at a faster rate.

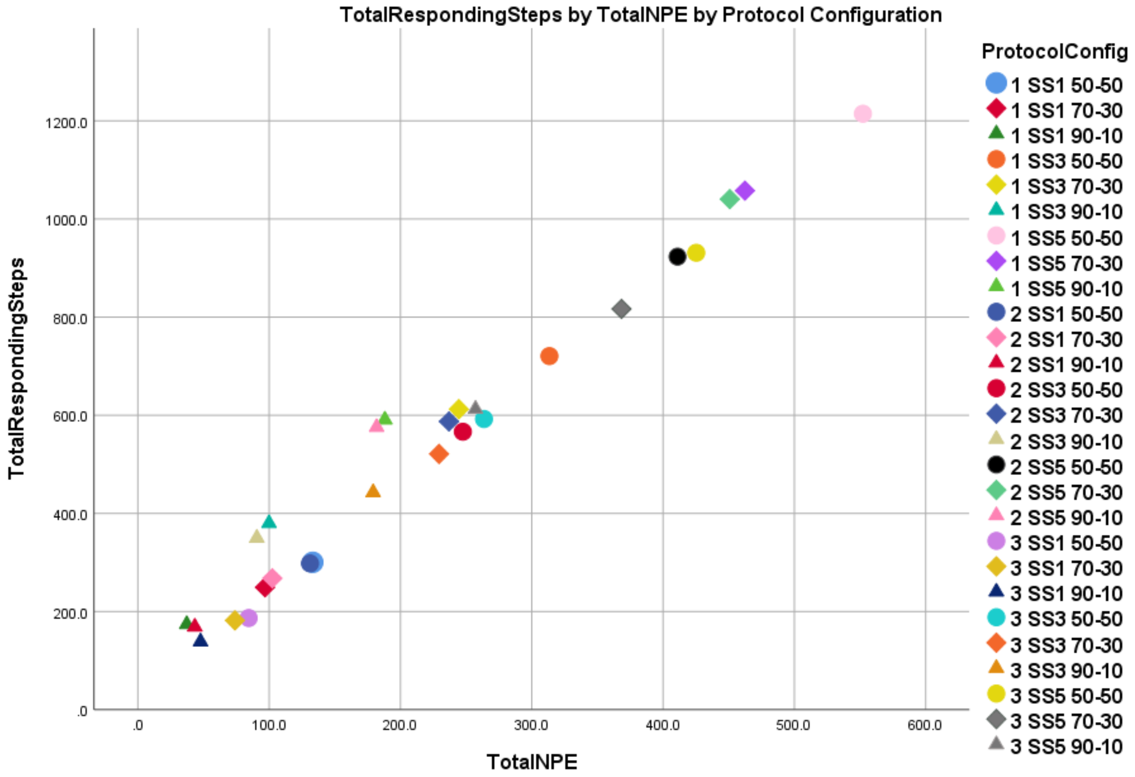


Figure 20. Scatterplot showing how many Responding Steps result in Non-productive effort.

CONCLUSIONS

The lower the signal strength, the more help messages are created due to the reduced reach. A poor signal range can result in no one responding to a message, the item is then abandoned by the initial robot that discovered it. When it is rediscovered by another robot a duplicate message is sent. A lower signal range may temporarily save battery power, but it does not necessarily improve overall efficiency as more messages may subsequently be sent if no robots responded to the initial message.

As expected, when the swarm is equally split with a 50%-50% composition the amount of message traffic is lower than is observed with an unequal swarm. The response rate to the help messages is heavily influenced by the signal strength, the greater the message range the more likely it is that the message will be answered. This also reduces the amount of wasted communication that occurs when a message doesn't receive a response. If an item is being continuously rediscovered, duplicate messages will be sent. If intra-swarm communication is low-cost then this is not a major concern, however the reduced communication has a knock-on effect that negatively impacts step count. The total simulation time is increased when the signal strength is set to one, the longer the simulation time the higher the step count. For an extreme swarm configuration of 90%-10%, Protocol 3 gives the most dramatic difference in results when comparing the lowest and highest signal strengths. The lowest signal range doubled the time and step count, in this case the power saved by using low range communication does not necessarily save overall battery life, as more time and effort is required to analyse all items.

More communication decreases battery life, as does responding to a help message, an Autonomic Manager would need to balance the trade-off between operations that consume a lot of power. This AM's decision-making process would be continuously updated as a result of the mission data that is transmitted from the swarm. Non-productive effort increases as the intra-swarm co-operation increases. While NPE wastes battery power, a lower amount of co-operation increases the time taken for all items to be analysed. A balance needs to be found between the amount of co-operation used in order to sustain the swarm's battery power for the mission duration, an AM could dynamically adjust the swarm to suit the circumstances.

REFERENCES

1. Hinchey, M.G., Sterritt, R., Rouff, C.A., Rash, J.L. and Truszkowski, W. 2006. “Swarm-based Space Exploration,” *ERCIM News (Emergent Computing)*, Vol. 64, European Research Consortium for Informatics and Mathematics, pp 26-27, January 2006.
2. IBM Corporation, “An Architectural Blueprint for Autonomic Computing,” April 2003.
3. Mohan, Y. and Ponnambalam, S.G. 2009. “An extensive review of research in swarm robotics,” *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India.
4. Sterritt, R. and Hinchey, M.G. 2005. “From Here to Autonomicity : Self-Managing Agents and the Biological Metaphors that Inspire Them,” *Proceedings of the 8th International Conference on Integrated Design & Process Technology Symposium (IDPT 2005)*, Soc. for Design and Process Science pp. 143–150.
5. Ducatelle, F., Di Caro, G.A., Pinciroli, C. and Gambardella, L.M. 2011. “Self-organized cooperation between robotic swarms,” *Journal of Swarm Intelligence.*, Vol. 5, No. 2, pp. 73–96, March 2011.
6. Curtis, S.A., Rilee. M.L., Clark., P. and Marr. G.C. 2003. “Use of Swarm Intelligence in Spacecraft Constellations for the Resource Exploration of the Asteroid Belt,” *Proceedings 3rd International Workshop on Satellite Constellations and Formation Flying*, Pisa, Italy.
7. Truszkowski, W., Hinchey, M., Rash, J. and Rouff, C. 2004. “NASA’s Swarm Missions: The challenge of Building Autonomous Software,” in *IT Professional*, Vol. 6, No. 5, pp. 47–52, Sept-Oct 2004.
8. Saunders, C., Sterritt, R. and Wilkie, G. 2016. “Autonomic Cooperation Strategies for Robot Swarms” *Proceedings of the 8th International Conference on Adaptive and Self-Adaptive Systems and Applications*, Rome, Italy.
9. Hinchey, M., Rash, J. and Truszkowski, W., Rouff, C.A. and Sterritt, R. 2005. “Autonomous and Autonomic Swarms,” *Proceedings of Software Engineering Research and Practice (SERP’05)*, Las Vegas, NV.
10. Rouff, C. “Intelligence in future NASA swarm-based missions,” 2007. *In AAAI Fall Symposium: Regarding the Intelligence in Distributed Intelligent Systems*, pp. 112–115.
11. Pan, Y., Rammig, F., Schmeck, H. and Solar, M. 2006. “Biologically Inspired Cooperative Computing” *IFIP 19th World Computer Congress, TC 10: 1st IFIP International Conference on Biologically Inspired Computing*, Santiago, Chile.
12. Sterritt, R. and Hinchey, M. 2010. “SPAACE IV: Self-Properties For An Autonomous & Autonomic Computing Environment–Part IV A Newish Hope,” *Proceedings of the 7th IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, Oxford, England.