MOM: a Tanker-based Foraging Framework

Adrian Kwok
School of Computing Science
Simon Fraser University

Abstract

In this paper, we describe a simple framework utilizing a large autonomous mobile storage and recharging tanker to increase the foraging efficiency of multiple smaller worker robots. This framework is targeted at scenarios where resources of interest are placed in size-constrained areas, requiring the use of low-energy miniscule robots for effective retrieval. We implement this framework through the Stage robot simulation environment, and perform a variety of experiments demonstrating the framework's increased efficiency, both in terms of speed and energy, and its other merits in open environments. Finally, we discuss possible extensions to the framework, including the possibility of incorporating previous work in the literature to improve its performance even further.

1. Introduction

The removal of radioactive materials in human-populated areas is an important problem due to its implications on health and public safety. Dangerous events such as the Chernobyl and Fukushima nuclear power plant disasters require the quick identification and removal of radioactive objects in wide areas; while larger objects can be identified and destroyed by humans with a handheld Geiger counter, smaller and even miniscule contaminated objects may require more sophisticated solutions. In this vein, we attempt to utilize robots to find and neutralize these small radioactive materials, and, for the purposes of this paper, we assume that the disaster area is similar to that of a forest clearing with no obstacles and radioactive materials randomly strewn across the clearing.

There are two particularly noteworthy properties associated with this task: firstly, we want to neutralize radioactive materials, and we can accomplish this by refining the radioactive materials into usable energy for our robots -- unfortunately, doing so requires relatively heavy and large machinery[7[. Secondly, a forest clearing implies that it is possible for radioactive materials to reside in small nooks and crannies – such as hollow tree trunks and badger holes – requiring the precision of smaller and agile robots. In this paper, we attempt to build a robot framework that exploits both of these properties to neutralize radioactive materials in order to solve the problem in question.

2. Motivation

There has been considerable work in the literature on multi-agent resource foraging tasks[1, 2, 3], including those that only rely only on local knowledge through the use of stigmergic[4] techniques such as simulated ant pheromones[5,6]. However, the focus of this paper is not on improving the effectiveness of the foraging algorithms themselves; our intent is to provide a simple yet robust, effective, and efficient foraging paradigm for which some of this work can be adapted to and for which future work can be built upon.

In modern day society, humans routinely create and utilize tools to improve the efficiency and efficacy of everyday tasks. With simplistic gathering tasks, most humans utilize a storage container to minimize the amount of travel needed to deposit multiple items, whether it be a basket or a push-cart as in the case with grocery shopping. In more complicated multi-agent collection tasks, such as garbage disposal services in urban cities, several foraging workers utilize a centralized mobile storage container – for example, a human-driven garbage truck – to increase item collection speed, to minimize overall energy consumption, and to increase area coverage; with the garbage truck analogy, a truck can go much further on a full tank of gas than a single garbage man can walk on a full stomach of food. This paper aims to mimic this behaviour to solve the previously described foraging problem through the use of a large mobile robot, nicknamed "MOM", which acts as a refueling station – similar to work done by Silverman et. al[10] – and a deposit basin for smaller simple scavenger robots, nicknamed "KID"s.

3. Framework Overview and Controller Design

In a conventional foraging system, worker robots routinely have to retrace their steps to refuel and to deposit resources, and more often than not these paths lead them through already explored areas, wasting valuable time that could be instead spent foraging. We attempt to lessen the amount of redundant travel that these worker robots have to do through the use of our framework; in our framework, MOM is a tanker robot which attempts to continuously reposition itself efficiently such that it is close to resources of interest. Each MOM robot has one docking port for every KID, situated on the sides of the robot, allowing KIDs to recharge their batteries by drawing from MOM's battery pack and to transfer foraged resources into MOM. For the purposes of this paper, we utilize one MOM and four KIDs.

In order for the framework to be completely autonomous, there is a fixed refueling station situated on every map which only MOM can recharge from; whenever MOM is low on battery, it heads back to this refueling station to recharge. Similarly, KIDs head back to MOM to recharge whenever they are low on battery; thus, this single refueling station acts as the main energy source for all robots in the framework, both directly and indirectly. As a special property of our predefined problem instance, MOM is also able to draw energy from gathered resources, prolonging the time for which it can do work before it has to head back to recharge.

Our inter-robot communication scheme is purposely simple: MOM does not keep track of the position of any of its KIDs, and KIDs only rely on MOM's global position and not that of its KID siblings. KIDs send messages back to MOM when they have seen or picked up a resource of interest, and as such the only information that is shared both ways is a list containing all resources that still need to be picked up. MOM keeps track of this list and updates it accordingly whenever it receives such messages from KIDs, and broadcasts this list to all KIDs on list updates; KIDs then use this list to aid their work. This communication scheme allows for minimal processing and data broadcasting by KIDs, which is ideal as they are meant to be low-energy robots.

The added complexity associated with a mobile tanker requires robust controllers that attempt to minimize or eliminate collisions from occurring when KIDs attempt to dock and undock from MOM; with a frequently moving tanker robot and worker robots which wish to dock onto the tanker, it is inherently difficult to ensure that such issues do not occur. Furthermore, we have the added requirement of complete system autonomy – for the purposes of this simple framework we cannot allow any robot to completely discharge its battery pack, which is a difficult problem as it may lead to unwanted situations where a KID is attempting to recharge from a MOM that is inadvertently heading in the opposite direction, resulting in wasted energy. We were mindful of these possible problems throughout the development of our robot controllers, and our specific design decisions will now be elaborated in detail for both MOM and KIDs.

3.1 The MOM Tanker Robot

The MOM tanker robot is given the responsibilities of both acting as a mobile deposit box for radioactive materials, and as a charging station for other robots. As a special case of the general scenario described previously, MOM is also capable of refining deposited radioactive materials into usable energy, through the use of something akin to a Stirling Radioisotope Generator[7], albeit slowly. Since the original defined problem is to dispose of all radioactive materials, we assume for the purposes of this project that by refining these materials into reusable energy, all radioactivity inherent in the material itself will be neutralized, and thus the original intent is fulfilled even if these fully drained materials are reintroduced into the environment.

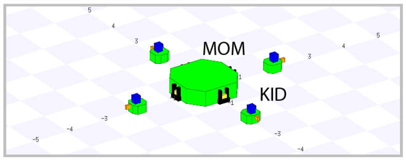


Figure 1: MOM and KID robots simulated in Stage 4.0.1

MOM itself is simulated in software in Stage 4.0.1[8], and is modeled after the Pioneer 2-DX robot[9], with a width and length of 1.5m x 1.5m, and a significantly enlarged battery capacity such that it can hold enough of a charge to refuel other worker robots potentially multiple times. As a result, MOM weighs 100KG and holds a maximum charge of 2000kJ, compared to other 0.5m x 0.5m scaled Pioneer 2DX example robots simulated in Stage which weigh 23KG. When refining radioactive materials, MOM is able to trickle charge at a rate of 50J per simulated time step, and each radioactive material holds 10kJ of usable energy.

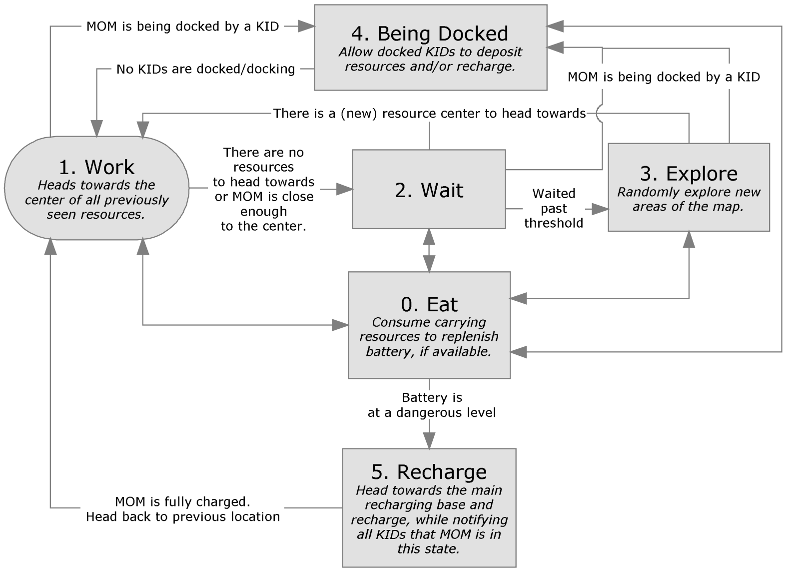


Figure 2: MOM flow diagram depicting modes and transitions

The state diagram conveying MOM's behaviour is shown in Figure 2. MOM's main goal is to be self sustaining – since all worker robots completely depend on MOM for sustenance, it must ensure that this goal be met for the entire system to be autonomous. Furthermore, it must attempt to position itself in an opportune location at all possible times such that it tries to lessen the amount of travel done by worker robots whenever depositing materials or recharging – if this is not achieved, then the whole framework is compromised, as it would likely perform no better than a static base with unnecessary added complexity. We will now discuss each of the states defined in Figure 2 in further detail.

3.1.0 MOM: Eat State

In the Eat state, MOM attempts to refine any radioactive materials that it currently is holding onto, and uses this to slowly replenish its battery; when a resource is completely refined – that is, it has lost all of its radioactivity – MOM will drop it back into the world. This state runs concurrently with all other states, aside from the Recharge state, and this is done deliberately so that refinable resources can be conserved for other states; it would be pointless to consume materials for energy when MOM is already heading towards a recharging station. If at any time MOM is low on battery, defined to be any remaining energy level lower than 60%, MOM will immediately enter the Recharge state, described in 3.1.5. During this transition to the Recharge state, a message is broadcast to all KIDs telling them that MOM is heading to recharge, so that all KIDs attempting to dock can stop trying to dock, and those that are currently docked can undock from MOM. Furthermore, MOM's current location is stored, such that it can return to it after recharging and resume whatever work it was doing previously.

3.1.1 MOM: Work State

In the Work state, MOM looks at its list of all currently tagged resource coordinates – that is, the Cartesian coordinates of resources that have been seen by worker robots, but have not been picked up yet – and calculates the position representing the center of all of these tagged resources. This list is updated whenever a KID finds or picks up a resource and notifies MOM via some communication method such as WiFi. MOM then heads towards the direction of this calculated position, and when it is close enough, stops and proceeds to the Wait state, described in 3.1.2.

If the list of all currently tagged resources is empty, MOM sits still and proceeds immediately to the Wait state. In the scenario where MOM is being requested to stop by a KID in order for the KID to dock, MOM immediately stays still and enters the Docked state, described in 3.1.4; this is to prevent KIDs from chasing MOM in order to dock, avoiding possible collisions and unnecessary energy waste as a result.

3.1.2 MOM: Wait State

In the Wait state, MOM simply sits still in an attempt to wait for a new location to head towards. If such a location exists – that is, the list of all currently tagged resources has a new center – it will proceed onwards to the Work state. However, if it has waited past a predefined threshold, MOM will enter the Explore state, described in 3.1.3 – this behaviour is to prevent stagnation, where the whole system is unable to explore any further than a subset of the map due to a lack of visible resources within the MOM and KIDs maximum scouting radius. The wait counter associated with this behaviour is not reset unless the list of tagged resources is updated with items, and as such, transitions from the Recharge and Docked states will immediately head back to the Explore state, if the list of tagged items is empty and unchanged.

As well, if MOM is being requested to stop by a KID in order for the KID to dock, MOM immediately will immediately enter the Docked state.

3.1.3 MOM: Explore State

In the Explore state, MOM attempts to explore the map, continuously heading in one direction for a fixed duration, before choosing a random new direction to head towards. Furthermore, when MOM collides with an object, it chooses a new random direction to head towards. This heuristic represents an extremely simplistic and naïve exploration method which still eventually covers all areas in an open environment within MOM's scouting radius, albeit likely inefficiently. As will be discussed in the Future Work section, there are many possible improvements that can be made with this algorithm such that MOM avoids retracing previously explored locations.

If at any time the list of tagged resources is updated with items, MOM will leave this state and transition to the Work state. Furthermore, if MOM is being requested to stop by a KID in order for the KID to dock, MOM immediately will immediately enter the Docked state.

3.1.4 MOM: Docked State

In the Docked state, MOM simply sits still while waiting for all docking and docked KIDs to finish their business, comprising of either depositing radioactive materials into MOM or recharging itself via MOM's battery. If there are no KIDs in the process of docking or are docked, MOM will then resume work by transitioning to the Work state.

3.1.5 MOM: Recharge State

In the Recharge state, MOM heads toward the main recharging station while avoiding obstacles and KIDs along the way. Once it has reached the recharging station, it replenishes its batteries until it is full, heads back to its previous recorded location as described in 3.1.0, and transitions to the Work state.

3.2 The KID Worker Robots

The KID worker robots are simple low-energy foragers who explore the map in a random manner in an attempt to find and transfer resources to MOM. When a KID is low on battery, defined to be 60% in this paper, it retreats to MOM and recharges its battery.

Each KID is simulated in Stage 4.0.1 as 0.5m x 0.5m Pioneer 2-DX robots, each weighing 23KG and having a battery with a maximum charge of 80kJ. To minimize collisions while undocking from MOM, each KID is outfitted with a low FOV rear laser, activated only when undocking. As discussed previously, KIDs are purposely small and simplistic in behaviour, such that it can navigate tight crevasses and minimize energy consumption. The controller design for KIDs is shown in Figure 3, and each state and associated transitions will now be discussed in detail.

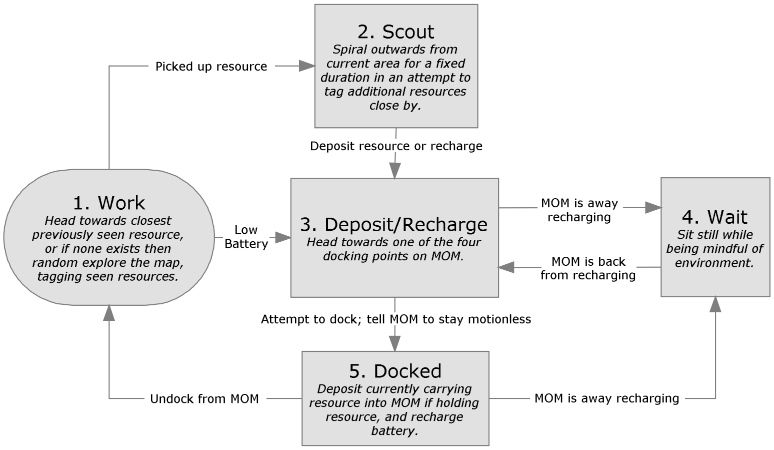


Figure 3: KID flow diagram depicting modes and transitions

3.2.1 KID: Work State

In the Work state, a KID looks at its list of previously tagged resources, synchronized periodically via a broadcast message sent out by MOM, as discussed previously. If this list is non-empty, the KID heads towards the closest resource in the list and once it is close enough it picks up the resource and proceeds to the Scout state, described in 3.2.2. If the list of previously tagged resource is empty, the KID explores the map randomly by heading in one direction for a fixed duration before heading in another random direction.

If at any given time the KID is able to see a resource that has not been tagged yet, it tags the resource by notifying MOM of this new resource, who then propagates this update to all KIDs in the system. This coincides with the behaviour discussed in the previous paragraph, as the KID will then immediately head towards this newly tagged resource as it will be the closest resource yet seen.

Lastly, if at any given time the KID is low on battery, defined to be any charge level lower than 60% in this paper, it will immediately transition to the Deposit/Recharge state, described in 3.2.3.

3.2.2 KID: Scout State

In the Scout state, a KID who is holding onto a resource will spiral outwards from the location of the resource it just picked up, in an attempt to efficiently explore the surrounding area and hopefully tag additional resources nearby. If the KID is low on battery, or if it has scouted the surround area past some fixed duration, it will proceed onwards to the Deposit/Recharge state.

3.2.3 KID: Deposit/Recharge State

In the Deposit/Recharge state, a KID who is either holding on to a resource or is low on battery will attempt to head towards MOM in order to dock. To do so, it heads towards the closest of the four docking entrances surrounding MOM, calculated via MOM's position and orientation and defined as the 0.5m x 0.5m area immediately 1m away from an actual docking port. If the closest docking entrance is occupied by another KID, it then cycles through the other docking areas until it finds one that is vacant. The KID then rotates such that its fiducial sensor is aligned with MOM's docking port, and notifies MOM to stop moving, via some nearfield communication protocol, and attempts to dock. Once the KID has successfully docked, it transitions to the Docked state, described in 3.2.5.

If at any time the KID is notified that MOM needs to go back to the home base to recharge, the KID will either transition to the Wait state if it is not in the process of docking – otherwise, it will attempt to halt its docking procedure and will back up a safe distance, and then transition to the Wait state.

3.2.4 KID: Wait State

In the Wait state, a KID stays motionless while waiting for MOM to finish recharging and return to its original location. If an object is about to collide with KID while it is in this state, it will attempt to avoid the object by moving out of the way. Once MOM has returned to its original location and resumed work, the KID will transition back to the Deposit/Recharge state.

3.2.5 KID: Docked State

In the Docked state, a KID deposits a resource into MOM if it is currently holding on to a resource, and recharges its battery to full if necessary. Once this is accomplished, the KID will attempt to undock from MOM by reversing until it is far enough away from the port, and will then transition to the Work state.

If MOM needs to recharge while KIDs are docked, all KIDs will attempt to undock to a safe distance away from its associated docking port, such that MOM will have an escape route if all four KIDs are surrounding it. Once undocked, the KIDs will then transition to the Wait state.

4. Experimental Results

To test our MOM and KID design, we simulated each robot and conducted a suite of experiments using Stage 4.0.1, with the primary aim of showing the effectiveness of our system in increasing foraging speed, improving energy efficiency, and improving area coverage. For each experiment, we utilize a simple box-shaped arena with no obstacles aside from four walls, and resources of interest are generated at the start of each experiment and are randomly and uniformly distributed throughout the arena. The starting position for MOM and its recharging base varies from experiment to experiment – in the “base at center” scenario, MOM starts at the center of the map. Similarly, a “base at corner” scenario would have MOM and its recharging base three meters away from the bottom left corner of the map. Figure 4 depicts a typical experiment starting instance.

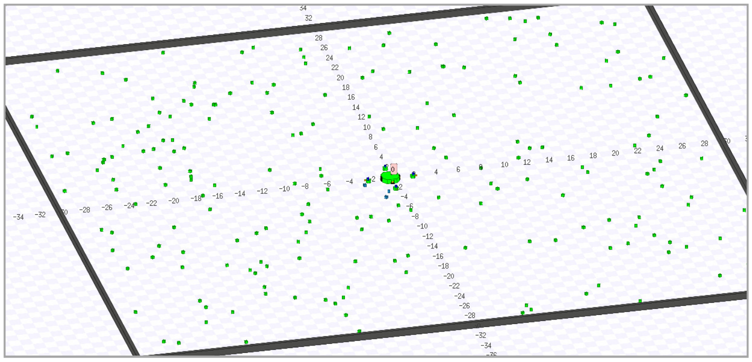
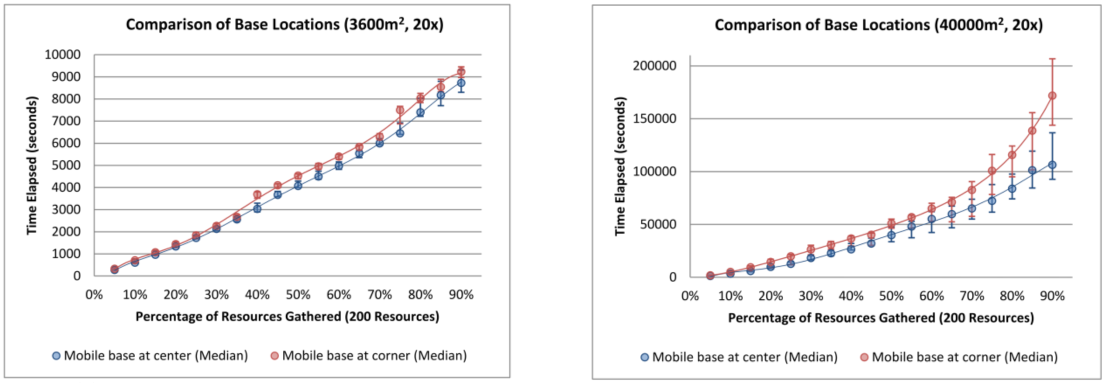


Figure 4: A typical experiment starting instance, with MOM at center

To ensure the validity of our results, we run and gather results for each experiment 20 times. Although infrequent, an experiment run is automatically stopped and its results are discarded whenever a robot collides with an obstacle or runs out of energy – this is to ensure fairness across a variety of different experiment scenarios and eliminates the possibility of erroneous results.

4.1 System Starting Locations

In order to confirm the validity of our experimental methodology for future sections, we first compare the differences in overall foraging speed due to different starting positions. The results are shown in Figure 5.



(a) 60m x 60m map, 20 iterations each

(a) 200m x 200m map, 20 iterations each

Figure 5: Comparison of starting positions and their effects on foraging speed

In both comparison tests, we analyse the amount of time it takes to deposit a set percentage of all strewn resources across the map. As shown in Figure 5(a), having MOM, KIDs, and MOM's recharging base at the absolute center of the map as opposed to one of the four corners in the map results in marginally better performance. The performance differences are greatly magnified when the same tests are run on a significantly larger map, shown in Figure 5b.).

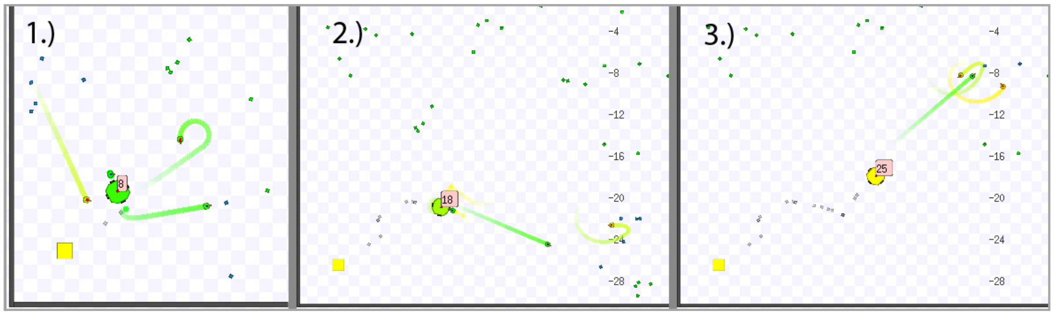


Figure 6: Observed resource "chaining" behaviour with the base in the corner scenario

These findings were initially unexpected due to the observed "chaining" behaviour of KIDs, shown in Figure 6 – when a KID picks up a resource of interest, it would spiral in place in an attempt to tag additional resources close by, and thus MOM would head towards these tagged resources. In an ideal situation, the KIDs would continually tag multiple resources in the same vicinity, such that MOM can move in a straightforward path without retracing its steps. By fixing the starting position to one of the four corners in the map, one would expect that two obstructing walls would bias the direction for which KIDs would move, and more likely result in this chaining behaviour. Conversely, if the starting position is in the center of the map, the KIDs would be equally likely to move in all four directions, minimizing the effectiveness of MOM, and thus the initial hypothesis was that it would be more effective to utilize a corner as a starting position.

As the results show, these assumptions were clearly misinformed or incorrect. After analyzing the behaviour further, we found that this likely has more to do with the observation that by restricting the initial directionality of all KIDs to one quadrant instead of four, some of the KIDs may be doing redundant work that could be accomplished with just a single KID heading in that direction. In the center case, it is more likely for both MOM and KIDs to spiral outwards from the center, resulting in much more efficient coverage over further distances. Due to these preliminary findings, we fix the starting location of MOMs and KIDs, as well as MOM's recharging base, to be the center of the map in all subsequent experiments.

4.2 Improvements to Foraging Speed

To analyse the possible improvements in speed – that is, the time required to collect a specific percentage of all resources – we use an immobile MOM as a baseline control. In the immobile scenario, MOM sits on top of the recharging base, and whenever a KID docks to refuel it automatically draws energy from the recharging base instead of from MOM's battery; MOM simply acts as an energy relaying system. Furthermore, whenever MOM receives a resource deposit, it does not refine it – however, this detail is irrelevant for these experiments as it would not affect the total energy consumed by the system. Under Stage 4.0.1, MOM consumes minimal energy in this static scenario as it is immobile and its laser and fiducial sensors are disabled.

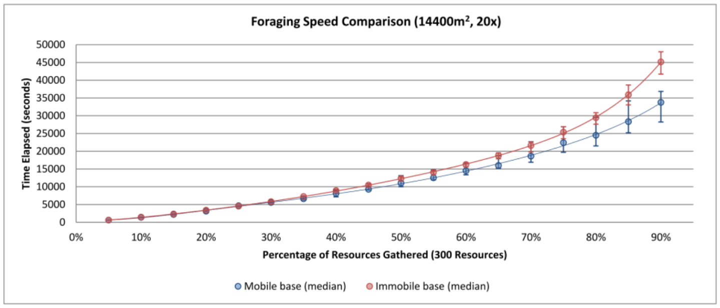


Figure 7: Comparison of foraging speed in a 120m x 120m open environment

The results of 20 iterations of both the mobile MOM and immobile MOM scenario are shown in Figure 7. The variance is higher with the mobile base as the system is more susceptible to poor random exploration paths chosen by MOM – if MOM is unlucky it may keep wandering in an already explored area much longer than preferred, stagnating the potential work of all four KIDs; in comparison, a KID's poor path finding decision in the immobile case only affects its own foraging performance and not that of its other three siblings. Regardless, by comparing median values, we can see that KIDs are able to forage 90% of the resources in the map 34.5% faster with a mobile MOM than with just a static base. This increase in foraging performance relates to our naïve heuristic where MOM attempts to situate itself in already-explored areas by heading towards recently tagged resources; the time spent by KIDs re-traversing already explored areas is time wasted not foraging, and this heuristic cuts down on this waste.

A noteworthy point to discuss is that the differences in foraging performance between mobile and immobile bases seem negligible until the KIDs need to explore the outskirts of the map. We can explain this in a mathematical manner – with an immobile base, we have KIDs moving in a random direction away from the origin, and then either retreating back when it finds a resource or when it runs out of battery. Clearly, this is analogous to rays emanating from (0,0), and the maximum area of coverage for any particular KID represents a near-perfect circle. Now, if we assume that there is one specific resource lying on the rim of this circle, then the probability that any particular KID will see this resource is roughly proportional to the circumference of the circle divided by the area that a KID can see with its sensors; thus, when the size of the foraged resource circle increases, the resources on the outer edge will become less likely to be seen during a KIDs trajectory if MOM is immobile. This problem is not as pressing when MOM is mobile, as the idea of an "outer edge" of the foraged resource circle becomes less relevant with a moving origin.

4.3 Improvements to Energy Efficiency

Utilizing the same test runs as in 4.2, we analyse the amount of total energy consumed by all MOMs and KIDs in the system, both in the immobile base and mobile MOM scenarios, and the results are shown in Figure 8. Since it takes longer for the immobile base scenario to completely forage 90% of the resources in our experimental map, we normalize the global energy consumed by each system divided by time to give us the number of watts consumed in order to forage each set percentage of resources, which is a fairer means of comparison.

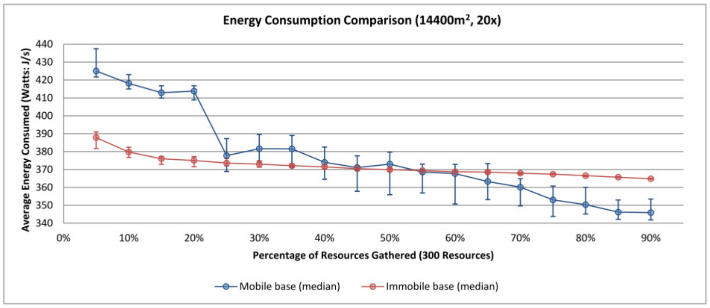


Figure 8: Power consumption comparison in a 120m x 120m open environment

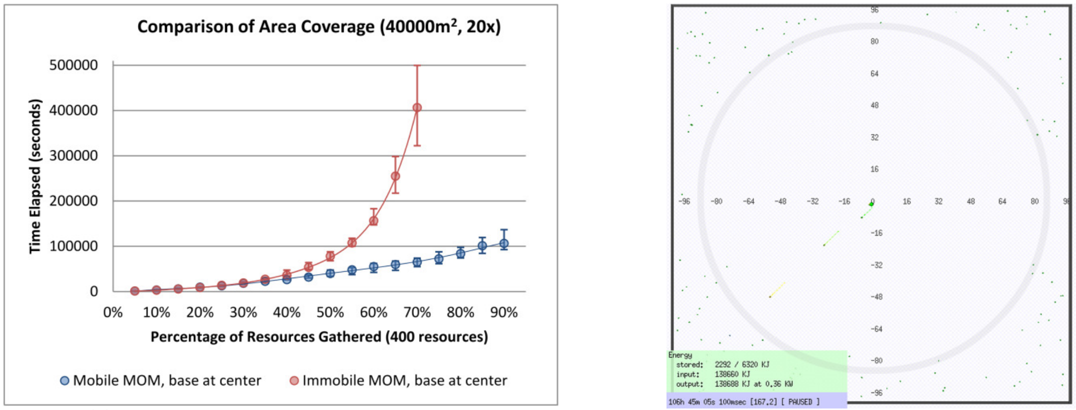
From these results, we can see that the mobile base consumes 5.2% less power (energy per second) to forage 90% of the resources in our test map – this is likely since, as demonstrated in 4.2, it is able to complete this task 34.5% faster. However, what is more noteworthy is that this is not true for all foraging target percentages: the overhead associated with MOM moving around initially dominates the power consumed by the system. This is likely due to MOM's movement heuristic: it only moves towards tagged resources unless the system stagnates, and there are a considerable number of resources being tagged when the system first starts, causing MOM to move more frequently.

It is interesting to note that in both scenarios that there is a steep decrease in power consumption during the first 15% of resources gathered – this likely has to do with the KIDs' sensors consuming energy while docked, yet not doing any work; there is initially is a large abundance of resources, thus KIDs will be constantly docking until resources become more scarce. Similarly, there are *increases* in power consumption sometimes in the mobile MOM scenario; this likely has to do with the movement associated with MOM retreating to the recharging station to refuel.

4.4 Improvements to Area Coverage

One of the most noteworthy and initially unexpected improvements by using this framework is the ability for energy-constrained foragers to explore much further locations as opposed to using a static recharging station. With a static recharging station, the maximum distance that each KID can travel is proportional to the energy that can be stored in its battery pack; in order for it to stay autonomous, a KID must head back to refuel while its energy is no lower than 50%, or risk dying. However, with a mobile MOM, the furthest distance that a KID can travel is now constrained by the furthest distance that MOM can travel plus the furthest distance that the KID can travel in the static case, which can be considerably further since MOM, by design, is meant to hold a significantly longer lasting battery.

To empirically demonstrate this feature of our framework, we utilized a very large open map, and the results are shown in Figure 9(a). We first ran several immobile tests for over 200 simulated hours (720,000 seconds), and found that the KIDs would forage no more than 70% of the resources on the map; we then used this value as a stopping point for subsequent experimental runs. A snapshot of the map with 70% coverage is also shown in Figure 9(b), and shows the bounded circular coverage of KIDs as described in 4.2.



```
Energy
  stored:   2292 / 6320 KJ
  input:    138660 KJ
  output:   138688 KJ at 0.36 KW
106h 45m 05s 100msec [167.2] [ PAUSED ]
```

(a) 200m x 200m map, 20 iterations each

(b) Static base in a 200m x 200m map after 106 hours

Figure 9: Comparison of area coverage in a 200m x 200m open environment

In the immobile base scenario, improving the foraging distance of KIDs while maintaining its necessary agile trait is inherently difficult; to improve this distance requires adding more batteries to a KID, which would make it larger and potentially unable to navigate the tight areas required by the original problem definition. However, adding such batteries to MOM and making it larger – perhaps, in height – does not pose as much of an issue, so long as MOM can still position itself in near-opportune places even in spite of this increased size. Furthermore, there may be possible extensions to MOM – namely refuel-chaining – that would allow for theoretically unbounded distances, and this is discussed in our Future Work section.

5. Future Work and Conclusion

In this paper, we described a basic framework utilizing a simple tanker robot to improve the performance of simplistic behaving worker robots. While the results presented are encouraging, there are still a considerable number of improvements that can be made for even more pronounced performance increases and for the framework to be completely feasible in practice.

With regards to map exploration, almost all of the exploration behaviour associated with both MOMs and KIDs are based on randomized walks. When KIDs explore and when MOM wanders to prevent exploration stagnation, there is no mechanism to prevent or avoid them from revisiting areas that they have already explored – this results in unnecessary wasted energy. It may be possible to take cues from ant-based algorithms to come up with a localized method for KIDs and MOMs to be aware of areas that they have already explored; for example, we can draw inspiration from the idea of breadcrumbs described by Vaughan et al.[6] and use similar droppings as markers for areas of which KIDs and MOMs should avoid.

One of the most important aspects of this framework lies with MOM repositioning itself periodically so that it can attempt to lessen the amount of travel required between a KID, a possible resource of interest, and itself. The heuristic described in this paper is simple but makes strong assumptions about the spatial locality of resources; MOM heads towards resources that have been recently tagged and thus, if there are more resources surrounding those that are tagged, this heuristic will be effective. However, if say there is only one resource that is tagged, and its general vicinity is empty, then MOM will have wasted a significant amount of effort getting to an area that is pointless to explore. Clearly, there are improvements to be made on this heuristic.

While a significant portion of the framework focuses on map exploration and minimizing the distance required for KIDs to travel, the recharging behaviour of both MOM and KIDs is also an important factor in overall system performance. In this paper we adopted a very conservative recharging method for MOM and KIDs: whenever their batteries reach a certain threshold, they immediately attempt to refuel so that that they don't die. However, this may not be the most efficient refueling method to utilize in all cases – for example, if a KID is clearly in a very good exploration area, it may be more beneficial for it to keep exploring, and, if it runs extremely low on energy, it can wait until MOM gets closer to it before it refuels. Similarly, if MOM is low on energy, it may be in MOM's best interest to stay where it is and wait until KIDs deposit more refinable resources into it instead of retreating a long distance to refuel. Lastly, with the current framework, once a KID has run out of fuel it is dead forever: it may be better to allow them to take risks with their energy and instead have MOM approach and refuel them if they are dead, as is done by Zebrowski and Vaughan[11]. In all of these possibilities, there is a clear balance between risk and reward that needs to be addressed more rigorously.

The final, and potentially the most important work that needs to be done in order for this framework to be feasible in practice is to adapt MOMs and KIDs to handle difficult terrain. The original proposed problem's real world counterpart would *not* be consist of completely flat terrain, and certainly would not be without any obstacles. This would require, at the very least, MOMs and KIDs to produce smart localization schemes, possibly utilizing complex plans so that they are able to reach one another efficiently. Furthermore, while KIDs are agile by design, a large and clunky robot such as MOM may have difficulty situating itself in opportune areas which generate the benefits given by this framework. This is a very difficult problem to address, and would require significant thought beyond the scope of this paper.

6. References

[1] Arkin, R. C., Balch, T., and Nitz, E., "Communication of behavioral state in multi-agent retrieval tasks". In *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation* (ICRA'93), pages 588–594, 1993.

[2] Shell, D. A. and Matarić, M. J., "On foraging strategies for large-scale multi-robot systems". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2717–2723, 2006.

[3] Østergaard, E., Sukhatme, G., and Matarić, M., "Emergent bucket brigading: a simple mechanism for improving performance in multi-robot constrained-space foraging tasks". In *Proceedings of the fifth international conference on Autonomous agents*, pp. 29–30, 2001.

[4] Holland, O. and Melhuish, C., "Stigmergy, self-organization, and sorting in collective robotics". *Artificial Life*, 5(2):173–202, 1999.

[5] Payton, D., Daily, M., Estowski, R., Howard, M., and Lee, C., "Pheromone robotics". *Autonomous Robots*, 11(3):319–324, 2001.

[6] Vaughan, R.T., Stoy, K., Sukhatme, G.S., and Matarić, M.J.. "LOST: Localization-space Trails for Robot Teams". In *IEEE Transactions on Robotics and Automation* 18.5 (2002): 796-812. Print.

[7] U.S. Department of Energy. "Space Radioisotope Power Systems: Advanced Stirling Radioisotope Generator". Jan. 2011. Web. 9 Dec. 2011.

[8] Vaughan, R.T., "Massively multi-robot simulation in stage". *Swarm Intelligence*, 2(2):189–208, 2008.

[9] ActivMedia Robotics. "Pioneer 2/PeopleBot Operations Manual". 2001. Web. 9 Dec. 2011. <http://dangermouse.brynmawr.edu/papers/pioneerman.pdf>.

[10] Silverman, M.C., Nies, D., Jung, B., and Sukhatme, G.S., "Staying Alive: A Docking Station for Autonomous Robot Recharging," in *Proc. IEEE International Conference on Robotics and Automation* (ICRA), 2002.

[11] Zebrowski, P. and Vaughan, R.T., "Recharging robot teams: A tanker approach". In *Proceedings of the International Conference on Advanced Robotics* (ICAR), 2005.