Using Mobile Robotic Agents to Increase Service Availability and Extend Network Lifetime on WSRNs

Ipek Caliskanelli
Department of Computer Science
University of York
York, UK
Email: ic539@york.ac.uk

Leandro Soares Indrusiak
Department of Computer Science
University of York
York, UK
Email: leandro.indrusiak@york.ac.uk

***Abstract*—Wireless Sensor and Robot Networks (WSRNs) are heterogeneous collections of sensor nodes and robotic vehicles that communicate wirelessly. In the last decade many research studies have attempted to address the challenging trade-offs of Wireless Sensor Networks (WSNs) that arise due to their resource limitations, however, much work is still to be done. A recent trend is to merge different subclasses of cyber-physical systems together to achieve the desired performance goals by benefiting from their heterogeneity. This paper presents a bio-inspired robot guidance technique that is used to improve network coverage, increase service availability and minimise energy consumption of WSNs using robotic agents on vehicles. We explore the performance consequences of the *Pheromone Signalling-based Load Balancing (PS) Robot Guidance* on an abstract level simulator which provides a system perspective. The effectiveness of the algorithm is evaluated with different network topologies and investigated on various scenarios. Simulated experimental results on mesh and sparse topologies validate that robot guidance based on *PS* increases network lifetime.**

I. Introduction

Wireless Sensor Networks (WSNs) consist of sensor nodes that have limited energy supply, sensor devices, a short-range radio and on-board processing capabilities. As the hardware technologies improve the cost of the sensor nodes decreases. One of the easiest solutions to maintain the required high performance from WSNs is deploying an excessive number of sensor nodes but this has the side effect of high computational redundancy and network overheads. An alternative solution for managing the limited resources of WSNs more effectively is to use techniques for low power design with high quality of service (QoS) and low level of redundancy [1]. This approach has been applied in two ways, often referred to as *classic* and *distributed* resource scheduling techniques. Classic resource scheduling and load balancing techniques apply central control mechanisms that are often deployed statically. These techniques are poor and inefficient in term of managing network lifetime and service availability due to their lack of knowledge of the network and the difficulties on handling the distributed nature, the size and the complexity of the problem [2]. One way to cope with the energy limitations and processing capabilities of WSNs is to use a dynamic, decentralised approach that can handle the distributed nature, the size and the complexity of the problem without increasing the network redundancy either at runtime or on-demand. As a result, recent research has focused on utilising dynamic and distributed load balancing and scheduling techniques. Although these distributed techniques improve the network performance significantly, the biggest bottleneck remains to be the nodes' restricted hardware. A growing trend is to merge sensornet platforms with cooperative robots and to use a distributed resource scheduling technique for managing the robot/sensor platform in the most resource effective way [3]. Due to their hardware being less restrictive, robotic agents are able to manage a heavier workload than sensor nodes, and can be used to improve the WSN performance.

In our previous work [4], [5], we define *PS*, a pheromone signalling-based load balancing technique. *PS* takes inspiration from the communicative behaviour of bees to address the trade-off between service availability and network lifetime. In this paper, we propose extending *PS* by introducing additional network elements in the form of robotic vehicles. Robots are able to move to areas of the network that are overloaded or where sensors nodes are dying, in order to offer themselves as processing elements. However, making decisions about where to position themselves in the network to best increase the service availability and network lifetime is non-trivial.

The main goal of this research is to *effectively* guide the robots to increase the network coverage, which will directly increase the service availability and extend the network performance. *Effective* network coverage in this research is defined as achieving the highest service availability by moving less. To achieve the desired effective network coverage, we have extended our *PS* technique to guide robots towards the areas of the sensor field where the sensor nodes have run out of battery and are unable to provide service.

II. Background and Related Work

In our initial work, *PS* [4], [5], we present a dynamic load balancing technique that is applied at run time at the application layer. *PS* is inspired from the pheromone signalling

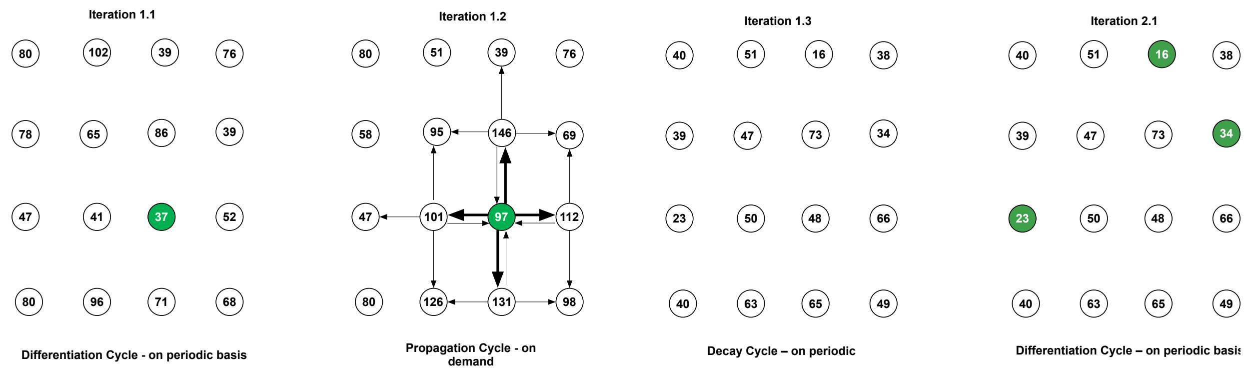


Fig. 1. Three important stages of PS algorithm: Differentiation, Propagation and Decay Cycles. *Green Indicates Queen Node

mechanism found in bees and provides distributed WSN control that uses local information only. *PS* is unique; unlike many load balancing approaches are applied at link or network layer [6], [7], [8] and balance only communication load, *PS* is an application-layer protocol and manages both computation and communication load. Briefly, *PS* is applied in three steps: a differentiation cycle, a propagation cycle and a decay cycle as shown in Fig. 1. The *differentiation cycle* takes place on each node of the network on a periodic basis at every T_{QN} time units. Each node decides whether or not to volunteer to be the node responsible for the task/event execution process: these are called *queen nodes* (QN). QN selection is based on each node's *pheromone level*: nodes who have a lower pheromone level than a pre-defined threshold become QNs. The second step of the algorithm is the *propagation cycle* and occurs on demand just after the differentiation cycle. When nodes become QNs, they start propagating some level of pheromone to the environment (nearby nodes) which indicates the resource usage in that part of the sensor field. A high pheromone level indicates high resource usage in that part of the network accordingly. The third step of the load balancing technique is called the *decay cycle*. This cycle occurs on a periodic basis on every node at every T_{DECAY} time units and it indicates the elapsed time in the environment. Natural pheromones disappear (decay) over time, and *PS* reflects this process by decreasing each node's pheromone level. In Fig. 1 nodes are represented with circles, and the numbers in the circles represents the pheromone level of each node. QN's are shaded green, and pheromone propagation is depicted using arrows. Darker arrows indicate a higher level of pheromone, whereas thinner arrows represent a lower level of pheromone. For a more detailed explanation of *PS*, see [4], [5].

To address our goal, this paper combines load balancing, task mapping, and network coverage issues on cyber-physical systems, applied in particular on sensornets and robots. Although load balancing, task mapping, and network coverage are different concepts, we merge these concepts in this paper and show an abstract level representation to provide a system perspective of WSRNs. We now present some significant research related to these three concepts.

Load balancing is often used on WSNs at different levels (MAC and network) to distribute network loads evenly. HEED [9], LEACH [6], and PEGASIS [10] are some of the well known examples of load balancing at the MAC level. On the other side, Heartbeat [7] and Gossip [11] protocols are good examples of network level load balancing. There are also existing bio-inpired load balancing protocols that are inspired by bees like Beehive [8], Beesensor [12] or by insect colonies like MONSOON [13].

One of the ways of applying load balancing is to use task mapping techniques in either dynamic or static manner. BTMS [14] and DNRS [15] are two of the significant techniques that have biological inspiration from zygote differentiation and artificial immune systems. Both BTMS and DNRS are applied dynamically (at run time) to optimise the network lifetime.

Network connectivity and coverage are widely researched topics that focus on how well the sensor field is monitored and tracked by sensors [16]. Approaches can be categorised into three classes [17]: area coverage, point coverage, and barrier coverage. Area coverage deals with the effective coverage the entire sensor field [18], [19], whereas solutions to point coverages deal with coverage issues between the set of selected points. Finally, solutions related to barrier coverage deal with minimising the probability of undetected penetration through the barrier [20]. Most of the existing research on network connectivity and coverage is developed at the lower levels of OSI stack and they deal with the lower level complexities of networks. Gage [21], however, defines system-level functionality of network coverage as the measure of the quality of system (QoS). Inspecting the network coverage from a system-level perspective without dealing with the low-level complexity of the network will bring the advantage of integrating heterogeneous network components in a complex system. Our research, therefore, focuses on the improving network coverage and connectivity of WSRNs from a system-level perspective using robotic agents on vehicles.

III. Problem Statement

The design objective of this research is to apply *PS* to guide robots effectively towards the areas of the sensor field where sensor nodes are not able to provide service. By developing the robot guidance based on *PS*, we increase the network coverage which also will increase the service availability whilst extending the network lifetime. We now define the performance metrics that we use throughout the paper to evaluate the effects of the robotic guidance.

1) *Service availability*: the number of services that are successfully completed divided by the total number of requested services within a period of time;
2) *Total energy consumption*: the sum of the communication and computation energy consumption within a period of time;
3) *Total distance taken by robots*: the total distance that a robot has travelled during the simulation.

In this paper, a *service* is defined as the composition of a number of inter-communicating tasks, and therefore a service is considered to be successfully *detected* only if all of its tasks are executed by at least one node. The minimum total distance travelled by the robots, combined with the highest service availability, indicates efficient guiding. We target, therefore, maximising service availability whilst minimising the total energy dissipation together with the shortest total distance taken by robots.

IV. Robot Guidance

This section explains the integration of robot guidance into the *PS* technique. In the *PS* technique, the level of pheromone indicates the resource usage in a particular area of the network. Areas in the sensor field that have lower level of pheromone at a given time demonstrate less resource usage as opposed to other parts of the network. Less resource usage may due to: (1) no events occurring in the neighbourhood; (2) events occur in the neighbourhood that are not in the detection range of particular node; or (3) nodes are already out of energy in that part of the sensor field. In our previous work, we apply our *PS* load balancing solution to cases (1) and (2), and show that by distributing the network load evenly we balance service availability and energy consumption. In this paper, we propose a robot guidance algorithm based on PS to solve case (3) by guiding robots into the areas where the sensor nodes are already out of energy. Incorporating additional robotic agents on vehicles and guiding the agents based on *PS* not only balances the network load (1),(2) but also improves the network coverage (3).

The three existing cycles of *PS* are explained in Section II as the differentiation, propagation and decay cycles. In addition to these three cycles that *PS* applies, we have extended *PS* with the robot behaviour to solve the given problem above (3). We now describe the robot behaviour algorithm and define how it occurs. The standard differentiation, propagation and decay cycles apply only on the sensor nodes, whereas the robot behaviour only occurs on robotic agents.

While the sensor nodes are deciding whether to provide a service or not to reduce the computational redundancy, it is essential to underline that the robotic agents are willing to provide service at all times to increase the network coverage. As with sensor nodes, robotic agents can receive pheromone from QNs if they are in communication range. Robotic agents act as QNs – they execute all tasks assigned to them. However, robotic agents do not propagate pheromone to other nodes in their communication range, so as to stop the robots interfering with the standard pheromone signalling mechanism. Listing 1 presents the robot behaviour in pseudocode. A robotic agent moves under two conditions: 1) if a robotic agent receives pheromone from a QN; 2) if the agent arrives at its destination without receiving any pheromone.

Listing 1. Pseudocode of the PS-based robot guidance algorithm.

```
if(pheromone received)
  PS−guided moving decision
else if (arrived at destination without
    receiving pheromone)
  randomly move
else
  broadcast communication link request
  establish local communication links
```

If a robotic agent receives pheromone it makes a moving decision and selects a target destination in the opposite direction of the received pheromone based on *PS*. The moving decision of the robotic agent is based on vector addition and its pseudo code appears in Listing 2. Given the mathematical formulation in the pseudocode and assuming all the network elements (sensor nodes and robotics agents) know their location as x and y coordinates, we calculate the angle of the received pheromone with the use of the sender's x and y coordinates. To do this, we resolve the horizontal and vertical components based on the amount of received pheromone level, h_i, and the coordinates of the sensor node. In order to find the magnitude, we sum up all the horizontal and vertical components. In order to determine the direction of the magnitude, we take arctangent of the magnitude and resolve x and y coordinates. This process happens on-demand as the robotic agents receive pheromone from QNs.

Listing 2. Robot Moving Decision

if $(h_i > 0)$
  for all the received pheromones (p) of the node
    $diff_X = p_{Sender_X} - currentCoordinate_X$
    $diff_Y = p_{Sender_Y} - currentCoordinate_Y$
    $\theta = ArcTangentQuadrant(diff_Y, diff_X)$
      $component_X = p.hd * \cos\theta$
      $component_Y = p.hd * \sin\theta$
      $Sum_X + = component_X$
      $Sum_Y + = component_Y$
$magnitude = \sqrt{Sum_X{}^2 + Sum_Y{}^2}$
$\theta_{destination} = ArcTangentQuadrant(Sum_Y, Sum_X)$
apply 180 degrees shift to $\theta_{destination}$
clear all received pheromones

If a robotic agent does not receive any pheromone by the time it arrives to its destination then the robotic agents picks a new destination at random and moves towards the new destination to increase the service availability by helping the sensor nodes. If a robotic agent does not receive any pheromone and has not yet arrived at its destination that means it is currently moving. In this case, the robotic agent continues to move towards its calculated destination whilst periodically broadcasting communication requests and updates its communication links nearby nodes.

V. Evaluation Environment and Experimental Results

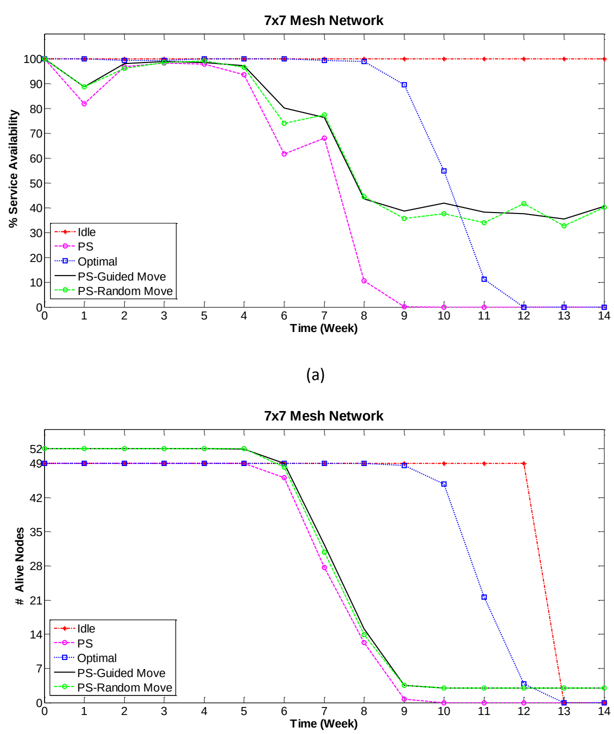
To evaluate PS, we have designed a three-tier system-level simulation model that represents the application-layer (consisting of tasks), platform layer (consisting of processing elements) and the mapper (that maps the tasks from the application-layer to the platform-layer). Our system-level simulator, *Fast*, is written in Java and it is an abstract simulator – trading accuracy for efficiency, scalability and flexibility. For further details about *Fast*, please see [4], [5]. In this research, we only extend platform model of our simulator (*Fast*), which consists of sensors, communication links, and robots.

This set of experimental work aims to evaluate the proposed PS-based robot guidance algorithm against three other scenarios:

1) *Idle* represents the absence of load: all nodes of the system do not dissipate any energy on computation or communication with the neighbours. It shows the maximum lifetime of the WSN (sensor nodes only) when they are not processing.
2) *Optimal* represents an artificial scenario for WSNs (sensor nodes only) to illustrate the highest service availability where each service is executed by only one service provider to ensure that no redundant processing takes place and minimum number of network resources is used.
3) *PS with Random Moving Robotic Agents* represents a WSRN where robotic agents move randomly in the sensor field without using PS to guide their movement. The decision to move is controlled by a given probability.

The performance of PS depends on a set of parameters that control the algorithm – specifying, among other things, the frequencies at which each cycle occurs. In [22] we present an approach to tune the parameters of PS for a given scenario. We use this technique to the parameters for this experiment. To ensure statistical significance, we have repeated each experiment 30 times. We compare each technique on a 7x7 mesh network and a sparse network with 70 nodes.

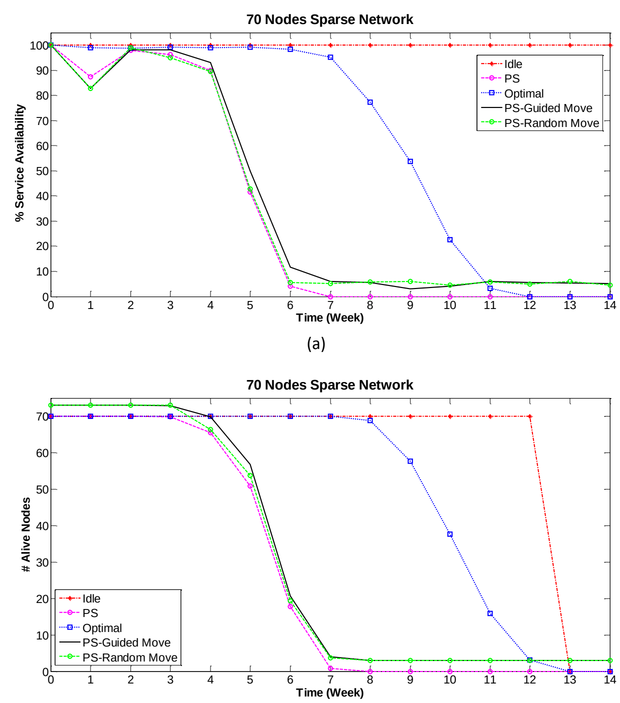
In Fig. 2 the percentage of the detected events and the number of alive nodes are shown for the 7x7 mesh network topology for *Idle*, *PS*, *Optimal*, *PS-Guided Move* and *PS-Random Move* scenarios. *Idle*, *PS* and *Optimal* scenarios are only applied on the 49 sensor nodes, whereas *PS-Guided Move* and *PS-Random Move* are applied both sensors and robots in the environment (49 sensors and three robots; 52 pieces of hardware device). *PS-Random Move* takes actions based on the given probabilities. As shown in Fig. 2 (a) and (b), the percentages of service availability and alive nodes are lowest in the *PS* scenario on WSNs (no robotic agents involved), and the highest is *Optimal* scenario. This means, using additional robotic agents on the WSNs is beneficial: robots do not interfere with the sensors, and improve network



(a)

(b)

Fig. 2. Experimental results: effects of *PS Robot Guidance* algorithm on 7x7 Mesh Network topologies showing (a) % Service Availability, (b) # Alive nodes.



(a)

(b)

Fig. 3. Experimental results: effects of *PS Robot Guidance* algorithm on 70 Nodes Sparse Network topology showing (a) % Service Availability, (b) # Alive nodes.

performance.

The *PS-Guided Move* scenario performs better, in terms of service availability, than the *PS-Random Move* scenario between week 4 and week 9 as this is the time that nodes begin to run out of energy. Starting from the week 10 there are no alive network nodes in the sensor field in any of the *PS*, *PS-Random Move* and *PS-Guided Move* scenarios, and as can be expected as there is no alive node, we do not observe any advantage of using *PS-Guided Move*.

Similarly, in Fig. 3, *Idle*, *PS*, *Optimal*, *PS-Guided Move* and *PS-Random Move* scenarios are illustrated on a sparse network with 70 nodes in a 30m x 30m grid as a test case [23]. The transmission range of network elements (both sensors and robots) are selected as 6m to establish a connected network. In order to specify the minimum transmission range that ensure connectivity, we use the topology control on sparse networks by Santi [24]. In Fig. 3 *PS-Guided Move* also outperforms *PS-Random Move* between week 4-9 and the difference between the two scenarios are higher than for the mesh network. This is due to the fact that the pheromone propagation is more effective in large scale networks and thus is why we claim that *PS* brings more advantages for large scale networks.

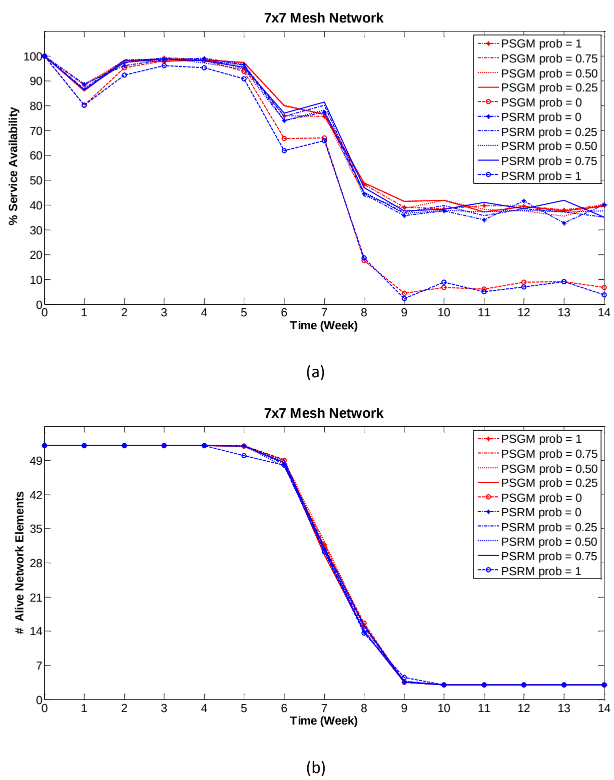


Fig. 4. Experimental results: effects of *PS Robot Guidance* algorithm on 7x7 Mesh Network topology showing various probabilities on move (a) % Service Availability, (b) # Alive nodes.

In Fig. 4 and Fig. 5, the effects of probabilities on deciding whether to move is illustrated for both the *PS-Guided Move* and *PS-Random Move* scenarios on mesh and sparse networks. 'PSGM' represents *PS-Guided Move*, whereas 'PSRM' represents *PS-Random Move* scenarios. Based on the given probabilities, each robot decides to move or not when the moving decision occurs. As we mention in Section IV, a robotic agent moves under two conditions: (1) if it receives some pheromone from a QN; or (2) if the agent arrives to its destination without receiving any pheromone. In *PS-Guided Move*, robots move based on *PS Guidance* when (1) occurs, and randomly move when (2) takes place. The probabilistic moves only occur when robots do not receive pheromone, when they are moving randomly.

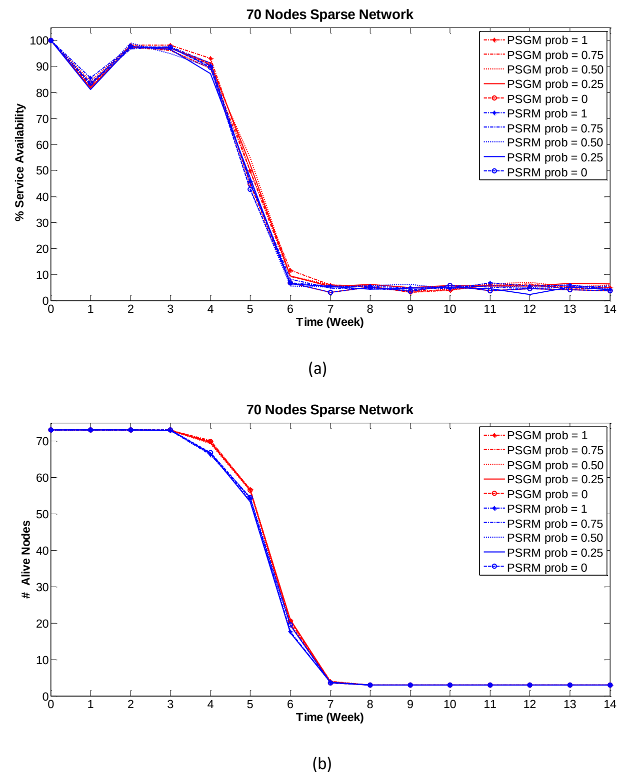


Fig. 5. Experimental results: effects of *PS Robot Guidance* algorithm on 70 Nodes Sparse Network topology showing various probabilities on move (a) % Service Availability, (b) # Alive nodes.

Given the probabilities on both scenarios and efficiency of the various probabilities in terms of service availability and number of alive nodes shown in Fig. 4 and Fig. 5, we can conclude that fixed robots (probability 0 = no move) brings very small benefit. On the other hand, when robots constantly move they achieve the highest level of service availability and lowest level of sensor energy consumption. It is an interesting point to notice that probabilities 0.25, 0.5 and 0.75 do not have much difference on service availability and sensor energy consumption, however, the total distance taken by robots are directly affected by the given probabilities. This then begs a question: is it better to sacrifice a little from the service availability and sensor energy usage but conserve the robots' energy by moving less? The answer of this question is based on the application domain and depends on how critical it is. By implementing *PS Guidance* we successfully direct robots into the areas of the sensor field where nodes are out of energy or much network activities occur. We will try to solve the trade-off between service availability and the total distance taken by a robot in the future as it is not the scope of this paper. However, we have done some basic analysis on the total distance travelled by a robot in Fig.6 to get an idea of the technique on the total distance taken.

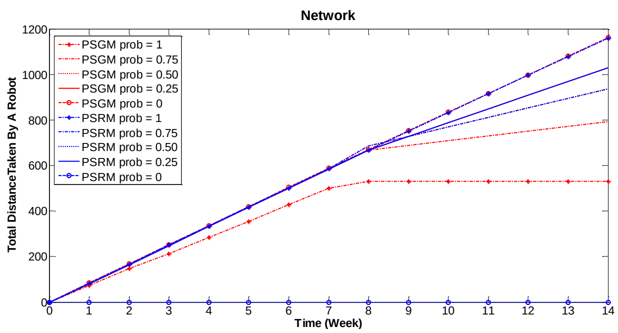


Fig. 6. Experimental results: effects of *PS Robot Guidance* algorithm on a network topology showing total distance taken by a robotic agent.

Fig. 6 illustrates the effects of the moving decision probability of *PS-Guided Move* and *PS-Random Move* on the total distance travelled by a robotic agent during the simulation. Due to the large number of experiments performed, each having 3 robots, we limit the results to the total distance taken by a single robot on a network. The numbers in this figure are not important as they change with the network topology. However, the behaviour of the scenarios and ratio of the results are similar to each other in all cases.

VI. Conclusion

In this paper, we have described an effective robot guidance technique that uses the *PS* load balancing algorithm to improve the network coverage in an attempt to address the trade-off between service availability and network lifetime. As the stationary sensor nodes are limited in processing capacity, we introduce mobile robotic agents in addition to the fixed sensornet topology. We propose to improve network coverage by guiding the robotic agents towards the areas of the sensor field where the sensor nodes are out of battery and are unable to provide service. Thus, this not only improves the network coverage but also increases the service availability and increases the network lifetime. Experimental results on mesh and sparse network topologies for various move-related probabilities demonstrate that our proposed *PS Robot Guidance* technique increases the service availability and extends the network lifetime as a result of improved network coverage, although the total distance travelled by the robots is large. In the future, we would like to consider the resource limitations of the robots, examining the tradeoff between the total distance taken by a robot and the total service availability of the network.

References

[1] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Communications, 2001. IEEE Int. Conf. on*, vol. 2, 2001, pp. 472 –476 vol.2.

[2] W. Li, F. C. Delicato, and A. Y. Zomaya, "Adaptive energy-efficient scheduling for hierarchical wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 3, pp. 33:1–33:34, Jun. 2013. [Online]. Available: http://doi.acm.org/10.1145/2480730.2480736

[3] X. Li, R. Falcon, A. Nayak, and I. Stojmenovic, "Servicing wireless sensor networks by mobile robots," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 147–154, July 2012.

[4] I. Caliskanelli, J. Harbin, L. Soares Indrusiak, P. Mitchell, F. Polack, and D. Chesmore, "Runtime optimisation in wsns for load balancing using pheromone signalling," in *NESEA, 2012. 3rd IEEE Int. Conf. on*, dec. 2012.

[5] ——, "Bio-inspired load balancing in large-scale wsns using pheromone signalling," *Int. Journal of Distributed Sensor Networks*, May 2013.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, vol. 2, Jan. 2000, p. 10.

[7] M. G. Gouda and T. M. McGuire, "Accelerated heartbeat protocols," in *18th Int. Conf. on Distributed Computing Systems*, 1998, pp. 202–209.

[8] H. Wedde, M. Farooq, and Y. Zhang, "Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior," in *Ant Colony Optimization and Swarm Intelligence*, ser. LNCS. Springer Berlin Heidelberg, 2004, vol. 3172, pp. 83–94.

[9] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct.-Dec. 2004.

[10] S. Lindsey and C. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *IEEE Aerospace Conf. Proc.*, vol. 3, 2002, pp. 1125–1130.

[11] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.

[12] M. Saleem and M. Farooq, "Beesensor: A bee-inspired power aware routing protocol for wireless sensor networks," in *Applications of Evolutionary Computing*, ser. LNCS, M. Giacobini, Ed. Springer Berlin Heidelberg, 2007, vol. 4448, pp. 81–90.

[13] P. Boonma and J. Suzuki, "MONSOON: A coevolutionary multiobjective adaptation framework for dynamic wireless sensor networks," in *Hawaii International Conference on System Sciences*, 2008, p. 497.

[14] Y. Hamouda and C. Phillips, "Biological task mapping and scheduling in wireless sensor networks," in *Communications Technology and Applications, 2009. IEEE Int. Conf. on*, oct. 2009, pp. 914 –919.

[15] B. Atakan and O. B. Akan, "Immune system based distributed node and rate selection in wireless sensor networks," in *Proc. BIONETICS*. ACM, 2006.

[16] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM international conf on WSNA*. San Diego, CA, USA: ACM, 2003, pp. 115–121.

[17] R. Mulligan and H. M. Ammari, "Coverage in wireless sensor networks: A survey." *Network Protocols & Algorithms*, vol. 2, no. 2, 2010.

[18] N. Aziz, A. Mohemmed, and M. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram," *ICNSC '09. Int. Conf. on*, pp. 602–607, Mar. 2009.

[19] Y.-R. Tsai, "Coverage-preserving routing protocols for randomly distributed wireless sensor networks," *Wireless Comm, IEEE Trans. on*, vol. 6, no. 4, pp. 1240–1245, Apr. 2007.

[20] Nojeong Heo and P. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," *Wireless Comm. and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1597–1602 vol.3, Mar. 2003.

[21] D. W. Gage, *Command control for many-robot systems*. Naval Command Control and Ocean Surveillance Center, San Diego, 1992.

[22] I. Caliskanelli and L. Indrusiak, "Search-based parameter tuning on application-level load balancing for distributed embedded systems," in *Proceedings of the 11th IEEE/IFIP EUC 2013*, 2013.

[23] P. Szczytowski, A. Khelil, A. Ali, and N. Suri, "TOM: topology oriented maintenance in sparse wireless sensor networks," *SECON, 2011 8th Annual IEEE Comm. Society Conf. on*, pp. 548–556, Jun. 2011.

[24] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.