TASK ALLOCATION IN DYNAMIC NETWORKS OF SATELLITES

Johannes van der Horst and Jason Noble
University of Southampton, Southampton, United Kingdom

ABSTRACT

The management of distributed satellite systems requires the coordination of a large number of heterogeneous spacecraft. Task allocation in such a system is complicated by limited communication and individual satellite dynamics. Previous work has shown that task allocation using a market-based mechanism can provide scalable and efficient management of static networks; in this paper we extend this work to determine the impact of dynamic topologies. We develop a Keplerian mobility model to describe the topology of the communication network over time. This movement model is then used in simulation to show that the task allocation mechanism does not show a significant decrease in effectiveness from the static case, reflecting the suitability distributed market-based control to the highly dynamic environment.

Key words: Distributed satellite systems, market-based control, task allocation, multi-agent systems.

1. INTRODUCTION

Distributed satellite systems offer numerous benefits over traditional monolithic satellite architectures, promising improved mission flexibility and robustness, as well as decreased cost and development time. Although these benefits are attractive, they come at the expense of increased system complexity as a large number of specialised units need to be coordinated. In this paper we envisage an incoming stream of multi-component jobs, e.g., requests to conduct earth observation, and we focus on one aspect of the coordination problem, namely task allocation. How do we decide which satellite should be assigned a task, given a dynamic network, heterogeneous individuals and limited information about network state? In previous work [9, 10] we developed a distributed task allocation mechanism and verified it in terms of efficiency, scalability and robustness to failure. In this paper we extend our analysis to take the orbital dynamics of a distributed satellite system into account.

We are specifically interested in using a groups of low-cost nano- and picosatellites, as we believe that distributed satellite systems can significantly increase mission possibilities for these systems. Satellites with different abilities and resources have to cooperate to complete mission objectives. We could, for example, have one class of satellite that specialises in earth observation and is therefore equipped with a camera, while the satellites that are responsible for communication with the ground station have high-power transmitters and enough memory to store data before downloading. However, the modest capabilities of the individual satellites mean that power is extremely limited, and the energy cost of communication will therefore have a major impact on the global system performance.

The combination of significant communication costs and the dynamic system topology makes this a hard problem, especially for large systems [5]. If we view the system as a network of computing agents, accurate task allocation requires the state of individual agents and their position in the network to be known. However, maintaining this knowledge requires communication. We can decrease the energy spent on communication by using only local inter-satellite links, but this in turn renders the network topology more sensitive to the relative positions of individual nodes which change as they orbit around the earth. Obtaining accurate information in this environment is expensive; instead we need a way of coping with imperfect knowledge. Instead of transforming the group of satellites into a stable orbital formation [7], our lack of fine-grained control forces us to find ways of managing the system, despite the above sources of uncertainty.

Although multi-spacecraft systems have been much studied from a control systems perspective, relatively little work has looked at their management, especially under dynamic conditions. The System F6 fractionated satellite project is pursuing a high-connectivity communication model, similar to the internet [2, 6]. This allows satellite nodes to share resources transparently, but implies significant communication and power requirements. Tripp and Palmer [8] addressed the problem of limited information by having the ground station communicate the state of the network to satellites. Satellites decide what to execute based on their own workload, the tasks that have been earmarked by other satellites for execution, and the tasks that were executed in duplication during the previous allocation round. Wu et al. [11] focused on optimising the routing strategy for a stable formation by using a multi-objective evolutionary algorithm, but this assumes knowledge of the positions of all satellites.

The distributed nature of the satellite problem led us to investigate techniques from distributed artificial intelligence, where the intelligence lies not in a powerful central authority, but rather in the relatively simple interactions between agents. Techniques from market-based control are particularly well suited to this scenario: local communication, heterogeneity and dynamic topologies are commonly encountered in human markets too. In the following section we describe our decentralised auction mechanism that balances the need to spread tasks across different agents against the need to minimise energy expenditure through localised allocation. By using task-centric routing, the allocation mechanism avoids the need for global knowledge at any point in the system. The Keplerian mobility model we use to capture the dynamics is described in Section 3; this model is then used in simulation to show that the task allocation mechanism is resilient to the changes in the network, and does not show a decrease in effectiveness from the static case.

2. TASK ALLOCATION

Tasks consist of multiple components, each of which is executed on a satellite with specific resources. The result of the completion of one component (e.g., image data from a camera) is transferred to the next satellite for the next step of task (e.g., fusing of multiple images). A task component is allocated with a reverse, sealed-bid auction mechanism, i.e., tasks are allocated to the cheapest bidder.

If a node has a task component to outsource, it assumes the role of auctioneer and announces to the task to its direct neighbours. They repeat the message to their neighbours, up to the packet time-to-live range, thus flooding the local network. The satellites that receive the auction announcement form the auction community. If a satellite in the auction community possesses the necessary resources, it submits a bid to the auctioneer. The bid value B is calculated using the size of the task z, the remaining energy e_{rem}, the maximum energy the node can store, plus the expected outsourcing cost, c_{os}:

$$B = z \left(\frac{e_{\text{max}}}{e_{\text{rem}}} + c_{\text{os}} \right) \quad (1)$$

Individuals that do a lot of work will deplete their energy, forcing them to submit higher bids, thus making under-utilised satellites more attractive to the auctioneer. The outsourcing cost is the estimated cost of outsourcing subsequent components of the task, and is calculated as a running average of previous outsourcing costs for similar tasks. The bid message is transferred back to the auctioneer by routing it along the same path that originally transferred the auction announcement. This is accomplished in a distributed manner by having a local routing table on every node that provides the next hop information. At every step along the way, the intermediate nodes aggregate all bids, and only retransmits the best (i.e., cheapest) bid towards the auctioneer. The transmitted bid is increased by a "commission" factor — this adds topological information to the bid value by making bids from nearby nodes less expensive than bids from distant nodes. If k is the commission and d_{bid} the distance between the auctioneer and bidders in hops, the value of the bid received by the auctioneer is given by:

$$B_{\text{rx}} = z \left(\frac{e_{\text{max}}}{e_{\text{rem}}} + c_{\text{os}} \right) (1+k)^{d_{\text{bid}}} \quad (2)$$

The auctioneer will therefore preferentially allocate to nearby nodes that are relatively under-utilised.

The use of bid aggregation distributes the calculation of the winning bids across several nodes; by limiting the number of packets transmitted, the allocation mechanism is made more efficient. The allocation process executes in parallel across the network, leading to efficient and adaptive task allocation at a global level. A message sequence diagram describing the task allocation mechanism is shown in Fig. 1, for a more detailed discussion of the allocation mechanism refer to van der Horst and Noble [9].

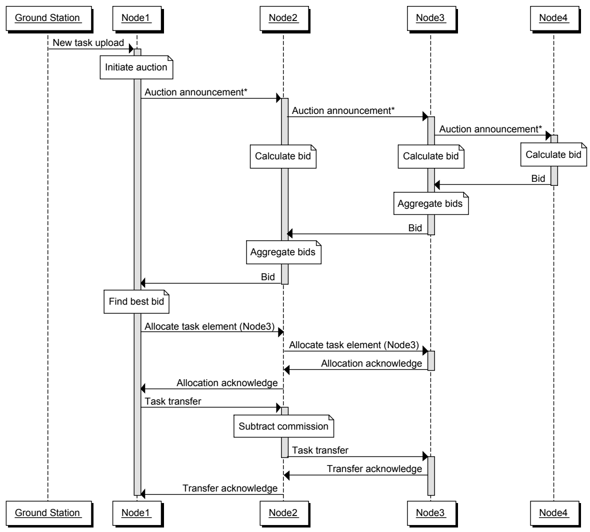


Figure 1. Message sequence diagram illustrating the allocation process for a single task component.

When allocating task components, we are interested in the capabilities and resources of the winning bidders, not their names. Instead of routing between two addresses, we can therefore use a task-centric routing scheme, where packets are routed according to the tasks they are associated with. To implement this, every node only needs to maintain a list of next-hop neighbours, indexed according to task identifier. The entries in this table are set up during the auction announcement and bidding phases, constructing a distributed route between auctioneer and successful bidder for the time required for allocation. Tasks are handled by their identifiers, instead of relying on the addresses of the satellites involved. This provides a fully distributed, adaptable routing layer for the allocation mechanism.

Markets are frequently associated with self-interested agents, intent on maximising their own welfare. However, in our system the agents work towards an efficient global state. The auction mechanism allows them to cope with having only local information; their utility functions have been designed to align their private motivations with the desired system-level performance.

3. KEPLERIAN MOBILITY MODEL

In previous work we compared the above distributed allocation mechanism to a centralised equivalent, to confirm efficiency, scalability and robustness to node failure [9, 10]. The scenarios investigated were, however, limited to simple, static network topologies. To extend our analysis to more realistic dynamic networks we require a representative mobility model for our application.

Mobility models are widely used in mobile ad hoc network (MANET) research to represent the dynamic behaviour of a communication network composed of mobile agents [1, 3]. As the agents move around, communication channels are formed between individuals in physical proximity to each other; these links are broken again if they move apart. The mobility model allows researchers to map the physical system to an abstract communication network that changes over time. Existing mobility models frequently rely on random movement to generate a dynamic environment. However, for a group of approximately co-orbiting satellites, individuals are subject to similar forces, with the variation in their orbital parameters determining their respective trajectories. As a result, we can expect a greater spatial correlation between satellites than can be expected for random movement. The interactions are also periodic: approximately the same formation occurs once per orbit.

We treat the distributed satellite system a group of satellites whose individual orbital parameters are derived from a reference orbit by adding a small amount of noise. The similarity in their parameters means the satellites are roughly co-orbiting, but that formation is not actively maintained. The small differences in individual orbits will cause the satellites to drift apart over time, which will require correction if the group is to stay connected.

The mobility model essentially solves the Keplerian equations to obtain the positions of all satellites at a specific point in time; the relative distances of between satellites are then used to find the adjacency matrix of the communication network. This adjacency matrix defines how nodes are connected to each other. It is convenient to use an imaginary point on the reference orbit to position the satellites around. This reference point orbits around the planet along with the satellites and serves as an origin for a local Cartesian reference frame. The use of this reference point is primarily a conceptual and visualisation aid, as it focuses the attention on the position of the satellites relative to each other, instead of the planet they are orbiting.

The steps required to determine the connectivity matrix at time t are the following[1]:

1. Calculate the Cartesian position u_{ref} of the reference point at time t
2. Calculate the Cartesian positions u_i of all satellites at time t
3. Centre the coordinate system around the reference point by translating all the satellite positions by $-u_{\text{ref}}$
4. Calculate the distances between all satellites to obtain the distance matrix.
5. Apply connection function to the distance matrix to find the adjacency matrix for the communication network.

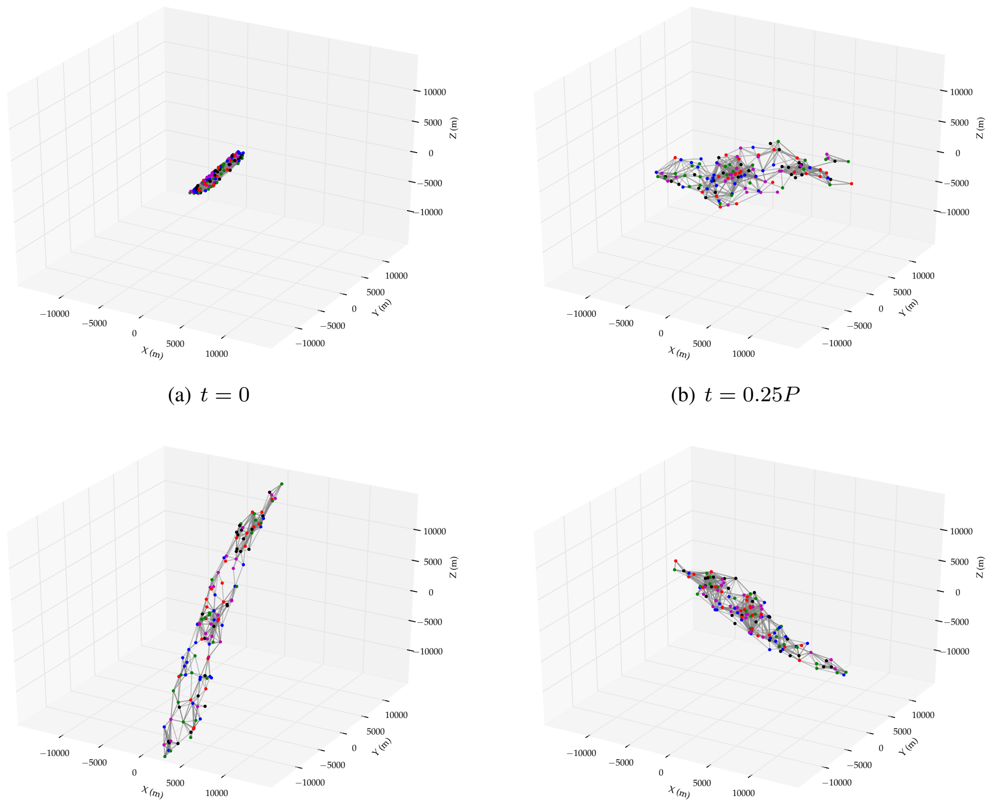
For a detailed discussion of the calculations required to perform the transformations, refer to Chobotov [4].

The connection function captures the propagation characteristics for the underlying communication medium. In this paper, we use a simple deterministic radio communication model: if two nodes are within a specified range, we assume they can communicate successfully. More realistic connection functions that incorporate noise and interference can of course also be used.

This model deliberately ignores non-Keplerian perturbations, such as geopotential or solar pressure. Although these forces have an important effect on the orbits of real satellites, we are primarily interested in the short term interactions between satellites, over time periods less than one orbit. The short-term effects of the perturbations are very small when compared to the Keplerian trajectories. The communication network dynamics, i.e., how frequently links are formed between satellites and for how long they persist, are therefore captured with sufficient accuracy.

The mobility model displays an oscillatory movement of satellites around the reference point. The exact behaviour is very much dependent on the orbital parameters, and variance thereof for individual satellites. The resulting local communication network displays continuous variation in topology and physical scale as spacecraft orbit around the earth, ranging from well connected to sparse and even disjointed. For specific parameter values that suppress the oscillatory terms, stable relationships between satellites can be found, but the dynamic connections dominate the communication network. When the communication range is small compared to the average inter-satellite distance, the network fragments. On the other hand, if the communication range is comparable to the diameter of the satellite cluster, a well connected network results.

[1] For optimised calculation of only the communication topology, steps 1 and 3 can safely be skipped.



(a) $t = 0$ (b) $t = 0.25P$

(c) $t = 0.5P$ (d) $t = 0.75P$

Figure 2. Network topology over the course of one orbit for a system of 125 satellites. Note the drastic change in topology, ranging from tightly clustered to sparse and elongated. The colour of nodes indicate specific satellites. P is the orbital period. The parameters used to generate these figures are the same as used for the simulation in Section 4.

In many cases the network topology ranges between these two extremes within one orbit, as is demonstrated by the network resulting from a slightly elliptical orbit in Fig. 2. This clearly shows how the satellite formation ranges from tightly clustered, as shown in Fig. 2(a), to being spread over a wide area (Fig. 2(c)). The local network around a node allocating a task changes continuously in terms of topology and composition, as shown in Fig. 3. To successfully allocate tasks, these changes need to be detected, or the network mapped at regular intervals.

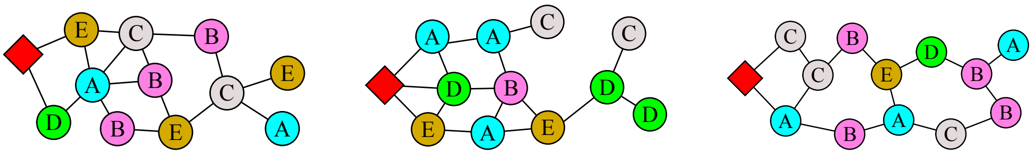
This mobility model provides the test case for verifying our task allocation mechanism, but it can also serve as a standalone mobility model for researchers in mobile ad hoc networking to test different management algorithms.

4. DYNAMIC PERFORMANCE

We are interested in maximising the task allocation and allocation efficiency of the system, because the more efficient the allocation mechanism is, the more energy can be spent on performing payload operations. For this experiment, we use networks where all tasks can be successfully allocated, but the energy overhead due to communication varies. By measuring the communication overhead, i.e., the total energy consumed by the system over time minus the energy spent on tasks, we can measure the impact of network dynamics.

To interpret our simulation results, we need a fair reference to compare it against. As the cost of allocation is strongly influenced by the network topology, we need to select a topology that provides a fair comparison, even though the dynamic network changes significantly over time. We therefore determine the topology of the dynamic network at a random time t, then use that as a static network for simulating allocation. If we repeat this for multiple t values, over a number of networks, the mean of the results should give a good indication of the performance without the effects caused by dynamics. This provides the static case.

For additional references we also measure the allocation overhead in two extreme cases: one when the satellites are spread out to the maximum extent to form the sparse case, and the connected case, where they are clustered together and well connected.



(a) t_0 (b) t_1 (c) t_2

Figure 3. Orbital dynamics result in continual change of the local neighbourhood of a node, both in terms of the types of nodes and the topology. The auctioneer is represented by the diamond, while nodes are coloured and labelled according to their types. If the network was static, the community would instead remain constant.

Table 1. Orbital parameters for simulated distributed satellite system. The reference orbit of the system has the parameters in the reference column, while the values for individual satellites are calculated by adding a uniformly distributed noise with the range shown in the error column.

Parameter	Reference	Error
Semimajor axis a	6878140	± 100 m
Eccentricity e	0.001	$\pm 10^{-6}$ rad
Inclination i	$\frac{\pi}{4}$	± 0.001 rad
RAAN Ω	0	± 0.001 rad
Argument of perigee ω	0	$\pm \frac{\pi}{4}$ rad
Initial true anomaly θ_0	$-\omega$	± 0.01 rad

4.1. Experimental setup

A slightly elliptical, 500km reference orbit is used to define the orbits of 125 satellites, using the orbital parameters in Table 1. The initial positions and orbits of the satellites are calculated by adding a uniformly distributed error to the parameters, as listed in the error column in Table 1. At $t = 0$, the satellites are clustered around the perigee of the reference orbit, but as they travel around the earth they spread out before clustering together again, as shown in Fig. 2. The connection function uses simple thresholding comparison: if two satellites are within 4000 m of each other, it is assumed that both parties can communicate with each other.

Five new tasks are introduced to system every 100 seconds, each task consists of five components that are executed sequentially by different types of satellites. The system is simulated for one orbit, during which 280 tasks (1400 task components) are allocated. Executing a task component uses 1 unit of energy, transmitting negotiation packets uses 0.005 units, while transferring a task between two nodes requires 0.5 units. Satellite energy regenerates at an average rate of 0.005 units per second, to represent the recharging of spacecraft batteries. The maximum energy that can be stored by any spacecraft is 10 units — careless allocation of tasks could lead to node exhaustion. These values represent a scenario where communication is cheaper than task execution, but the cumulative energy cost of communication forms a significant portion of the total energy expenditure.

The skills of the satellites are selected with a uniform probability from the set of 5 task component types. Every satellite has only one skill, so the system consists of approximately 25 satellites of each class. Note that both the composition of the network and the positions of individual satellites varied between runs.

The dynamic case uses the Keplerian movement model to modify the network topology over time. The static case uses 20 different time values to generate different topologies for every run. As the orbits of individual satellites are elliptical, the connected case is found by using the network at $t = 0$. The sparse topology is found halfway through the orbit, at $t = 2883$. Fifty runs were used to generate the results presented below.

4.2. Results

The mean energy used in allocation by the negotiation and transfer packets is shown in Fig. 4. The dynamic case required 1323 units of energy, while the static case required 1347 units, approximately the same amount. The sparse network used only 1164 units, while the well connected case required 1512 units. The standard error on the measurements ranges from 1.6 to 2.8. For comparison, task execution required 1400 units of energy. Approximately half the allocation energy is used for transferring tasks, while the remainder is required by the large number of negotiation packets.

The results show that the market-based task allocation mechanism is not adversely affected by the changing communication network, with a similar performance as the averaged static case. The extreme cases clearly demonstrate how the cost of allocation can vary within one orbit. The high cost associated with the well-connected case can be ascribed to the large auction community resulting from the small network diameter.

Tests using a different number of satellites, or placing them in other orbits show similar results, as long as the component satellites have approximately similar orbits.

5. DISCUSSION

The similarity in performance between the static and dynamic cases can be ascribed to the ad hoc nature of the

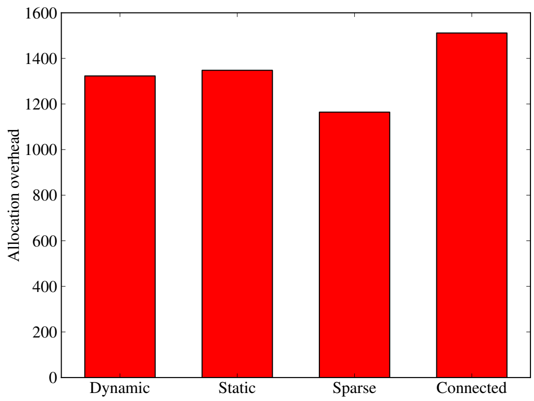


Figure 4. Mean allocation cost using dynamic, average static, sparse and well-connected topologies over one orbital period. The dynamic network requires approximately the same amount of energy to allocate as the average static case. Standard error on the measurements ranges from 1.6 to 6.2 (errors bars are not plotted due to their negligible size).

task allocation mechanism, as well as the differences in time scales between allocation and network changes. By holding an auction for every task component, the current state of the local network is always used to determine allocation. This approach is best suited to scenarios where the network changes more frequently than tasks are allocated — the cost of holding an auction should be less than the cost of incrementally tracking changes in the local community.

Despite the continuous movement of satellites, the average connection lifetime between two nodes in the communication network is significantly longer than the duration of an auction — on the order of hundreds of seconds for connections, versus seconds for auctions. From the point of view of the auctioneer, the local network is therefore effectively static during an auction, thus allowing successful allocation. If network changes were to disrupt an auction, the auctioneer can restart the auction to retry.

We have demonstrated that ad hoc, decentralised task allocation and task-centric routing allow us to successfully manage task allocation in distributed satellite systems with realistic network dynamics. The resilience of the allocation mechanism to a changing network topology implies that the requirements for fine-grained control of individual spacecraft can partially be addressed on a network management level: instead of accurate formation maintenance, coarse positioning of spacecraft to stay part of the network is sufficient. This makes the use of smaller, simpler spacecraft viable in distributed satellite scenarios.

The Keplerian mobility model presented in this paper also links distributed satellite systems more closely to existing work on MANETs. This opens the way for the testing of a number of MANET routing and control methodologies in distributed satellite systems. The high cost observed for the well-connected network suggests that the allocation mechanism can be made more efficient by using an adaptive time-to-live range. The periodic nature of topologies produced by the Keplerian mobility model also raises the question of whether the periodicity can be exploited to simplify management of the system.

REFERENCES

[1] Bai, F. and Helmy, A. (2004). A survey of mobility models in wireless ad hoc networks.

[2] Brown, O. C., Eremenko, P., and Collopy, P. D. (2009). Value-Centric design methodologies for fractionated spacecraft. In *AIAA SPACE 2009 Conference*.

[3] Camp, T., Boleng, J., and Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing*, 2.

[4] Chobotov, V. A., editor (2002). *Orbital Mechanics, Third Edition (AIAA Education Series)*. AIAA, 3rd edition.

[5] Durfee, E. (2004). Challenges to Scaling-Up agent coordination strategies. In Wagner, T. A., editor, *An Application Science for Multi-Agent Systems*, volume 10 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*.

[6] Lobosco, D. M., Cameron, G. E., Golding, R. A., and Wong, T. M. (2008). The Pleiades fractionated space system architecture and the future of national security space. In *AIAA SPACE 2008 Conference*.

[7] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R. (2004). A survey of spacecraft formation flying guidance and control. Part II: control. In *American Control Conference*, volume 4.

[8] Tripp, H. and Palmer, P. (2010). Stigmergy based behavioural coordination for satellite clusters. *Acta Astronautica*, 66(7-8).

[9] van der Horst, J. and Noble, J. (2010). Distributed and centralized task allocation: When and where to use them. In *Self-Adaptive Networks (SAN) Workshop, IEEE International Conference on Self-Adaptive and Self-Organising Systems (SASO2010)*.

[10] van der Horst, J., Noble, J., and Tatnall, A. (2009). Robustness of market-based task allocation in a distributed satellite system. In *Advances in Artificial Life: Tenth European Conference on Artificial Life*.

[11] Wu, X., Vladimirova, T., and Sidibeh, K. (2008). Signal routing in a satellite sensor network using optimisation algorithms. In *IEEE Aerospace Conference*.