JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 8, May 2011

Stigmergy for Nanosatellite Cluster Coordination without Intersatellite Links

Howard Tripp* and Phil Palmer†
University of Surrey, Surrey, GU2 7XH, United Kingdom

DOI: 10.2514/1.48212

A distributed coordination architecture based on the principles of stigmergy has previously been proposed as an efficient mechanism to coordinate spacecraft without the overhead of significant interswarm communication. This article looks at the emergent properties of the system as a whole based on local spacecraft behaviors, and how these behaviors can be adjusted to achieve mission goals. The significance of this approach is that it frees the ground station from the micromanagement of individual spacecraft (despite the lack of intersatellite links) which is a crucial requirement for the feasibility of future cluster/swarm missions. To understand the system fully, an analysis is presented for simple systems that have different characteristics. Performance is evaluated in terms of intuitive behaviors (greedy, considerate, proactive, and obstinate) that are mathematically defined. This reveals that performance can be steadily improved by making the behaviors more "self-aware." Simulations are then extended to swarms involving larger numbers of spacecraft to demonstrate scalability and more realistic scenarios for practicality. The final contribution of this work is to introduce various mechanisms for supervisors, and/or the ground station, to dynamically adjust behavior to achieve load balancing across the cluster.

I. Introduction

THERE is increasing interest within the space community to move toward multiple platform distributed missions. Distributed missions have many potential advantages such as signal separation (e.g. large synthetic apertures, where multiple sensors can be combined to create a virtual sensor whose capability would be impossible to replicate on a single platform), signal space coverage (e.g. multipoint sensing where multiple targets of interest can be viewed simultaneously), and signal combination (e.g. data fusion, where different types of data can be collected simultaneously for a larger range of measurements) [1]. Concept missions of the future also point toward missions involving larger numbers of smaller platforms working collaboratively for enhanced science return [2]. The elimination of single-point failures and continuous upgradability of the factorization paradigm [3] provide further engineering and economic advantages. However, there inherently becomes a point at which there are too many spacecraft for a single ground station to control, as is the current approach. Direct ground control also breaks down on longer range exploration missions such as missions to Mars and the asteroid belt. The only feasible solution is to shift more of the ground segment tasks up to increasingly intelligent autonomous spacecraft, hence reducing the costs and demands on the ground.

Autonomous spacecraft operation is still in its infancy with a lot of earlier work concentrated on the ability to modify and maintain dynamic operation plans for individual spacecraft. For example, the EO-1 Autonomous

Received 20 November 2009; revision received 1 February 2011; accepted for publication 14 February 2011

* Surrey Space Centre, Guildford, UK, h.tripp@surrey.ac.uk
† Surrey Space Centre, Guildford, UK, p.palmer@surrey.ac.uk

Science Agent [4], NASA's deep space one mission [5], or the NEAT architecture for scheduling [6] and low thrust trajectory calculations [7] both of which use an evolutionary algorithm on-board. When the autonomous collaboration of multiple spacecraft is considered, much of the work focuses on the astrodynamic problem of formation flying. Operational autonomy studies for multiple spacecraft is now receiving growing interest, the first major contribution of which related to the (defunct) NASA TechSat21 mission. Results from this work concluded that a hierarchical "team-based" control structure was the best trade-off between excessive communications and excessive computation [8]. This makes sense as many of the systems proposed require a complete world view and hence are not scalable to more than a few spacecraft due to the communications requirements [9]. Other approaches that focus on smaller numbers of spacecraft operating within tightly constrained limits include D-SpaCpanS [10] that shows the principles of adaptable and mobile managers that can reconfigure and autonomously organize hierarchies of control or the Shared Activity and Coordination model [11], where spacecraft negotiate with each other directly to partition the work amongst themselves. One of the inherent assumptions of multiple spacecraft autonomy is the need for intersatellite links [12] either for the direct negotiation between the spacecraft or just to share system state. Although distribution via clever and efficient protocols can be envisaged, the difficulty of maintaining intersatellite links is likely to be prohibitive to small spacecraft [13]. Clearly, though, multiple platform missions (using the small low-power spacecraft) are of interest; indeed NASA has established the Autonomous Nanotechnology Technology Swarm (ANTS) [14] concept mission as a common umbrella framework designed to stimulate emerging autonomous and autonomic systems technologies. Such large-scale distributed missions will require the use of swarm intelligence [15] of which stigmergy is a crucial component that has yet to be significantly researched for space applications.

This article therefore focuses on a system based around the principles of stigmergy. Stigmergy is an agent-based, behavioral coordination mechanism introduced by Grassé [16] inspired by natural living systems such as ant colonies and schools of fish. This approach endows the system with emergent properties that can causes self-organization [17] as is the case with ant food foraging and collective fish "navigation"—realistic models of which can be created using simple behaviors rules [18]—which allows for potential implementation on platforms that have limited capabilities such as nanosatellites.[‡] Coordination is achieved using infrequent communication via indirect modifications to a common (digital) environment [19] (rather than by direct message passing) with decisions based on local information [20]. Essentially, it is a compromise solution that combines the benefits of minimal intersatellite links (which are highly nontrivial in space) whilst critically freeing the ground station from the micromanagement of direct remote control. Further attractive features of the system are that it is inherently scalable to large numbers of spacecraft and, due to its simple behavioral algorithms, is able to run on the small low-power microspacecraft that are characteristic of the envisaged swarms.

The system previously proposed in [21] (Fig. 1) is hierarchical and "tree-like." Tasks to be performed by the cluster flow up from the ground station (root) along with some digital environment information and global performance metrics (goals). As the environment information and goals pass through the layers, they are slightly modified by intermediate nodes (supervisors) based on their own local information. For example these intermediate nodes may filter or aggregate the digital environment information, to "guide" other nodes, but the actual tasks remain unaltered. At leaf nodes, worker spacecraft perform tasks basing their choices on the environment information. They return completed tasks down the hierarchy along with some feedforward information about predicted future performance. Again this downstream information and tasks are modified and monitored by intermediary nodes. When this information reaches the ground station, it is subject to more processing before being looped back up, completing the cycle. Hence, as the task and environment information flows around this loop, it is modified by the various nodes that it passes through. Equally, this information is used to determine local behavior and actions, giving the two properties that are needed for the indirect coordination of stigmergy.

Essentially a spacecraft's "behavior" is characterized as a probability distribution over the task space (i.e. a probabilistic preference for particular tasks). This probability is defined using an orthogonal set of Chebyshev polynomials to ensure generality. Adjusting the weightings of the different polynomials controls the shape of the probability distribution (behavior) and a genetic algorithm is used to such the parameter space of these weightings. Specific

[‡] For example, power, CPU power, or data storage. In general, however, the point is that the design constraints are extremely tight and so on a coordination system with low system overheads is essential.

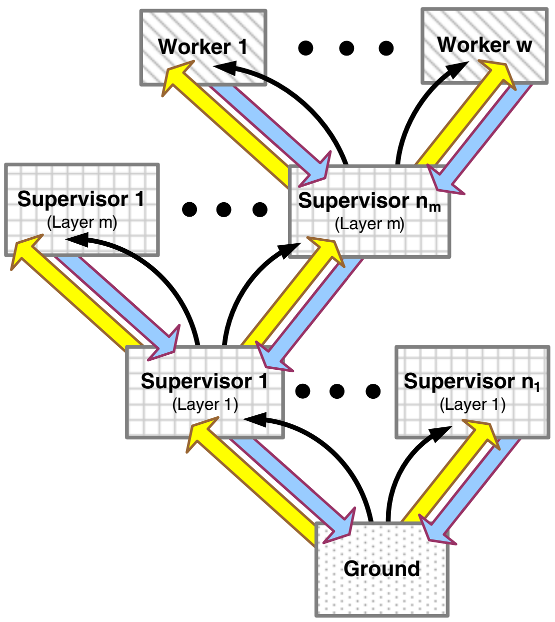

Fig. 1 Schematic of the information flow in the hierarchical stigmergy task allocation architecture.

consideration must also be given to the dynamic characteristics of the problem and the limited processing power of small microspacecraft when designing genetic algorithm operators. The detail of this on-board implementation strategy is outside the scope of this article and is detailed in previous work [22].

The rest of this article details some of the mathematics behind the environmental information flows (Section II) that are used by the system along with the characteristics of behaviors (Section III). Section IV explains some performance properties of the behavior characteristics on simplified scenarios. Section V shows how performance can be improved by increasing the intelligence (or self-awareness) of the spacecraft on more realistic scenarios. Section VI outlines the fundamental trade-offs that exist with the stigmergy system. Section VII introduces a more realistic model for resource as power and goes on to discuss the concept of load balancing among heterogeneous spacecraft, and finally some discussions and conclusions are presented in Sections VIII and IX.

II. Environment

In order for the system to operate in a stigmergetic manner it is necessary to have a common digital environment that is available to all the spacecraft. In the simplest case, this is achieved by the ground station RF broadcasting environment information at regular intervals. This is a very low overhead and can be represented as mathematical abstractions over the resource space. Optimization of the representations would also be considered. Hence, in practice, it is likely that the digital information overhead could be represented in only a few hundreds of bytes. As broadcast (and the environment information) is independent of the number of tasks or spacecraft, this is completely scalable communications architecture.

The spacecraft need to modify the environment, but rather than any kind of rebroadcasting architecture to propagate these changes (which would require intersatellite links) spacecraft simply respond directly to the ground station as and when they complete tasks and have the opportunity (as they do already, hence almost negligible impact). The ground station collects together all the data and then the information is periodically rebroadcast.[§] Issues surrounding the frequency with which the broadcasts occur (i.e. the communications bandwidth) are explored in [21].

[§] Clearly, the system can be extended and improved with intersatellite links that allow lower latency in maintaining and synchronizing the common environment, but such hybrid systems are left for future work.

To define the common environment in more detail, this article begins by assuming that the system contains a number of tasks each of which has an associated resource requirement (e.g. time, power, fuel, memory, etc). Tasks are considered to be independent of each other as scheduling constraints are enforced at the lower level [21]. The number of tasks at each resource point forms a (multidimensional) histogram in the resource space. For the purposes of this initial analysis, only a one-dimensional case is considered where each task has a single resource value, r (which is helpful to consider as "user task priority"). Without loss of generality, we restrict the resource (priority) to be a real value between 0 and 1 and hence we can define the task map at update t as follows:

$$\begin{aligned} T(t,r) &= \text{No. of tasks with resource requirement } r \\ \int_0^1 T(t,r)dr &= \text{Total no. of tasks in the system} \end{aligned} \tag{1}$$

The set of tasks is periodically broadcast[¶] to all the spacecraft so that the distribution of current tasks is always known. Spacecraft perform tasks independently, returning the results to the ground station, and therefore, in between task broadcasts, the task map will become out of date. Which tasks are performed by a spacecraft (and planned to be performed in the future) is governed by the spacecraft behavior which is a probability distribution over r. This current behavior can be reported to the ground station along with completed tasks with a very low overhead (it can be entirely described by a handful of numbers, i.e. the weighting functions for the set of orthogonal Chebyshev polynomials that sum to give the distribution). The ground station can sum these behaviors over all the spacecraft to give a snapshot view of which tasks the cluster as a whole is trying to perform. This summation function, $F(t,r)$ represents feedforward information that is collected together by the ground station and updated (i.e. broadcast along with the task map) every time step, again with very low overhead. In order to reduce susceptibility to noise and chaotic behavior, information from previous updates is also carried forward. This information is slowly decayed[#] over time so that historical information is gradually forgotten. The decay parameter is arbitrarily chosen to be a reasonable value of 0.8, which gives a recurrence relation on $F(t,r)$:

$$\begin{aligned} B_i(t,r) &= \text{Behaviour pdf of spacecraft } i \text{ at time } t \\ F(0,r) &= 0 \\ F(t,r) &= 0.8 \times F(t-1,r) + \sum_i B_i(t-1,r) \end{aligned} \tag{2}$$

Because each spacecraft works independently, multiple spacecraft may happen to perform the same task duplicating effort unnecessarily.[**] The ground station keeps a count of these duplications in a particular time step, $d(r)$, and in a manner analogous to the feedforward information, feedback (duplication) information, $D(r)$, is maintained over the time steps:

$$\begin{aligned} D(0,r) &= 0 \\ D(t,r) &= 0.8 \times D(t-1,r) + d(t-1,r) \end{aligned} \tag{3}$$

Again this feedback duplication information can be rebroadcasted to the cluster each update. As with the feedforward information and spacecraft behavior reporting, it can be described in a concise mathematical format. This format represents an extremely low message overhead to piggyback on top of the task updates that are already present. The environmental information flows described now ensure that each of the spacecraft has access to four crucial pieces

[¶] Or more likely just the changes from the previous update for bandwidth efficiency.
[#] Or evaporates if this feedforward information is considered using the pheromone analogy.
[**] If task duplication is actually desirable, then this could be achieved by introducing two (or more) distinct, but identical, tasks into the system so that the desired task duplication is achieved. Without loss of generality, therefore, it can be assumed that although two task's objectives maybe identical, they still remain logically distinct and so it is always undesirable to duplicate any specific task instance.

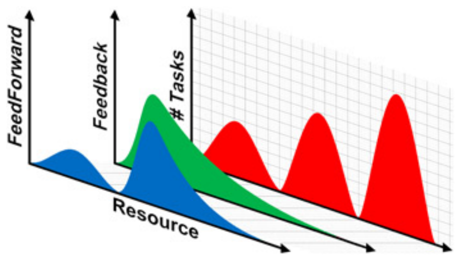

Fig. 2 Representation of the environment information available to the spacecraft showing the tasks along with the feedback and feedforward information across the same resource dimension.

of information (Fig. 2):

- Its own local behavior (and history of behavior), $B(t, r)$
- The global distribution of tasks currently in the system, $T(t, r)$
- The global predicted feedforward information, $F(t, r)$
- The global duplication feedback information, $D(t, r)$

III. Behaviors

The next step is to define what an actual behavior represents. In essence, it guides the task selection for a spacecraft into particular regions of the resource space (in probabilistic sense). A behavior might, for example, try to perform tasks with lower resource requirements, or try to avoid duplicating the effort. For this article, four different "characteristics" are chosen that can be used to make up behavior. These four characteristics are each chosen to explicitly make use of one of the four pieces of information just described above. By assigning a higher or lower degree of importance to each characteristic (possibly dynamically), the behavior of the spacecraft changes; and by choosing different sets of behaviors for the members of the team various levels of coordination and performance can be achieved. The four characteristics are outlined below along with their mathematical definitions. These functions are designed to return a normalized value in the range 0.0–1.0 such that 1.0 represents the optimal behavior for the particular characteristic.[††]

A. Greedy

This approach represents the selfish mode. It makes use of the task map information described in Eq. (1). A greedy spacecraft will try to do what is best for itself locally, with no consideration for others. Meaning it will attempt to perform the most valuable tasks currently in the system. Mathematically this is the (resource, r) expectation of behavior probability, B_i, multiplied by the taskmap, T. As Greedy is looking to minimize resource usage, the expectation is subtracted from 1.

$$G_i(t) = 1 - \frac{\int_0^1 rT(t, r)B_i(t, r)\, \mathrm{d}r}{\int_0^1 T(t, r)B_i(t, r)\, \mathrm{d}r} \tag{4}$$

B. Considerate

This is the feedback mode where significant emphasis is placed on the result of previous historical performance. This characteristic makes use of the feedback information (Eq. 3). The considerate spacecraft will avoid regions of previous overlap (i.e. multiple spacecraft redundantly performing the same task). In other words, it considerately performs tasks in which no other spacecraft is interested (and hence most likely of low value). Mathematically this is of the same form as Greedy, but instead of resource, r, expectation, Considerate uses, duplication, D, expectation. Unlike r, the bounds of D are not fixed between 0 and 1, hence an additional (max–min) normalization factor

[††] It should be noted that in this context the resource r follows the computer science standard with 0 being the "best", or in this case highest priority.

is required.

$$C_i(t) = 1 - \frac{\int_0^1 D(t,r)T(t,r)B_i(t,r)\,\mathrm{d}r}{[\max(D(t,r)) - \min(D(t,r))]\int_0^1 T(t,r)B_i(t,r)\,\mathrm{d}r} \tag{5}$$

C. Proactive

This is the feedforward mode, where emphasis is given to predictions of future behavior. The proactive spacecraft will attempt to avoid predicted regions of contention/overlap using the feedforward information (Eq. 18). In a sense this is also considerate, but it is proactive rather than reactive. Mathematically, this is of the same form as considerate but using feedforward, F, expectation rather than duplication, D, expectation.

$$P_i(t) = 1 - \frac{\int_0^1 F(t,r)T(t,r)B_i(t,r)\,\mathrm{d}r}{[\max(F(t,r)) - \min(F(t,r))]\int_0^1 T(t,r)B_i(t,r)\,\mathrm{d}r} \tag{6}$$

D. Obstinate

This is the ignorance or inertia mode where the spacecraft makes no use of any environment information and carries on regardless. An obstinate spacecraft will attempt to do exactly what it did in the past regardless. Mathematically, this is again of the same form as considerate, but using behavior history, B_{iH}, expectation rather than duplication, D, expectation. The added competition in this case is the need to build up the behavior history over the previous time steps. The historical behavior is decayed by 20% each time step, before being added to the current behavior. As this is already normalized, no additional normalization is needed.

$$O_i(t) = \frac{\int_0^1 B_{\mathrm{iH}}(t,r)T(t,r)B_i(t,r)\,\mathrm{d}r}{[\max(B_{iH}(t,r)) - \min(B_{\mathrm{iH}}(t,r))]\int_0^1 T(t,r)B_i(t,r)\,\mathrm{d}r} \tag{7}$$

where $B_{\mathrm{iH}}(0,r) = 0$ and $B_{\mathrm{iH}}(t,r) = 0.8 \times B_{\mathrm{iH}}(t-1,r) + B_i(t-1,r)$

E. Combined

How "good" a behavior is can now be determined mathematically using a weighted sum of the four characteristics (weightings g, c, p, and o, respectively):

$$M_i(t) = g.G_i(t) + c.C_i(t) + p.P_i(t) + o.O_i(t) \tag{8}$$

where $g, c, p, o \geqslant 0$ and $g + c + p + o = 1$.

By setting any of the weightings to 1 a "pure" behavior can be obtained. However, generally, a behavior is influenced by all the characteristics to some degree, usually with one dominating. This function (Eq. 1) depends on the task map, feedforward, and feedback environment information as well as the previous behavior history. As such, it is clear that a particular behavior can never be considered as good, only good in relation to a particular system state. Hence a behavior is not a fixed shape, but changes dynamically in response to the system so as to maintain its particular characteristics. This function provides a natural objective function that can be used to identify the optimal behavior autonomously on-board the spacecraft using a genetic algorithm search strategy (detailed in [22]).

The subject of discussion in this article is how teams of spacecraft with different behaviors (i.e. behaviors with different characteristics) affect the performance of this collective system as a whole. In essence, there are two things to be varied: (a) the weightings of the characteristics of the different behaviors and (b) the arrival pattern of tasks into the system. The feedback, feedforward, and behavior history distributions are all determined parameters of the system that result from the dynamic interactions of spacecraft behavior, and so cannot be varied explicitly. Of course, they then subsequently affect the behavior distributions in the dynamical system, and hence, predicting the emergent behavior of such a "swarm system" based on the behavior of individual agents is not generally analytically tractable. The results presented in this article are therefore based on multiagent simulation as is normally the case [18, 23].

IV. Simplified Scenarios

Before considering the details of simulation results, some general properties of the configuration of the simulator should be outlined:

1. The tasks are performed probabilistically based on the behavior preference to represent the lower level real-time and scheduling due to conflict/failure. Tasks are also performed at a fixed service rate as on average task throughput of spacecraft is invariant over their mission life (to a first approximation).
2. Each of the tasks has a single resource giving a one-dimensional, unit resource space with the dimension frequently referred to as *priority*. Upstream task, feedback and feedforward information updates are all received simultaneously by all spacecraft (to simulate broadcast from a ground station). This is a simplification of the more general multidimensional representation.
3. Downstream task completion and feedforward information is also performed simultaneously (for ease of simulation), but in practice, this is more likely to "trickle down" asynchronously. In this flat structure, such a simplification is not deemed to have any noticeable effect, as it is the broadcast of the new update that is important.

Equally, before the discussions of performance can begin, the concept of a specific performance metric also needs to be introduced. In the first case, this is considered to be task duplication—what percentage of the tasks performed by the cluster was entirely unnecessary.

$$\%\ \text{Duplication} = 1 - \frac{\text{No. of unique tasks performed}}{\sum_{\text{all spacecraft}} \text{No. of tasks performed}} \tag{9}$$

In these simulations, all tasks are required to be performed in the same amount of time. As there is a periodic update of new tasks into the system, each of the spacecraft performs an identical number of tasks every update, and the total number of tasks performed during the simulation is constant. The percentage duplication is therefore a direct proxy for task throughput—the higher the percentage duplication, the lower the number of unique tasks performed and the lower the rate at which the cluster can processes tasks through the system. Clearly, low % duplication is desirable.[‡‡]

To understand how a group of spacecraft can work collectively to perform tasks in this resource space, it is prudent to start with highly simplified examples. Collective operations naturally require a minimum of two spacecraft, which is the starting point here. To minimize the number of simulation variables, one of the spacecraft is set to a fixed constant behavior, in this case chosen to be purely greedy (Eq. 20). At all times, the spacecraft strives to perform the tasks that have the lowest resource requirement. The second spacecraft adopts a spectrum of behaviors that are a combination of two characteristics, greedy, and one of the others:

- 100% greedy to 100% considerate.
- 100% greedy to 100% proactive.
- 100% greedy to 100% obstinate.

A. Static Task Map

The initial task map configuration is referred to as the static task map. In this case, the tasks are uniformly distributed, $T(r) = 1$, and no new tasks are added into the system by the ground station at the update points. Hence all the dynamics are entirely determined by the stochastic nature of the behaviors. As there is no input into the system, it is impossible to reach a steady-state condition and therefore the performance values are taken at $t = 140$, which is when roughly 70% of the tasks have been completed. Although no new tasks are added, system updates are still required to transmit the feedback and feedforward environments information and also to indicate which tasks have been completed. This is the indirect communication between the two spacecraft. In this case, the rate is chosen to occur periodically with each spacecraft performing 25 tasks in between updates [24]. (So, for an Earth observation spacecraft in LEO orbit, taking two or three images per orbit, this would correspond to a roughly daily update.)

The static task map, although mainly useful for understanding behaviors and as a validation exercise, is a good model for situations where tasks are updated very infrequently, or perhaps where the majority of the work has been

[‡‡] If task duplication is really desired, this can be modeled by adding multiple identical tasks into the system.

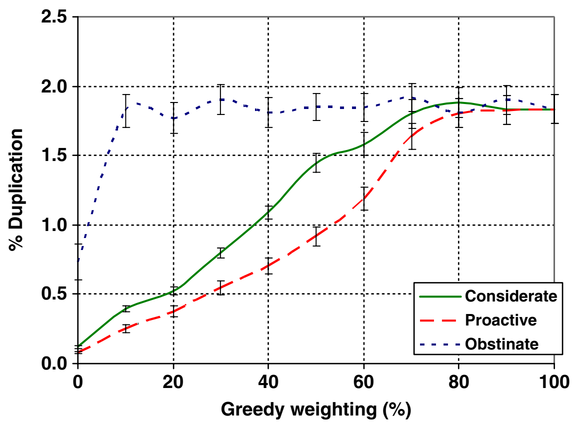

Fig. 3 Amount of task duplication generated by different behavior characteristics for a static task map. Error bars represent 95% confidence intervals.

preplanned. Such situations may occur on things such as science missions where the set of tasks and objectives is broadly known beforehand.

The first results (Fig. 3) that can be seen is that as the second spacecraft becomes greedier the amount of task duplication increases. This is a fundamental and intuitive result for this type of system—the more similar spacecraft behaviors are, the more they tend to perform in similar regions of the task space and hence replicate each other's task selection. In other words, the problem is more challenging, and performance degrades, when spacecraft are homogeneous rather than heterogeneous. Fig. 3 also gives some indication of the effects of different behavior characteristics. The obstinate behavior is the worst performing behavior, as it makes very little use of the information around it. The considerate behavior does a lot better by guiding the second spacecraft away from the contentious regions, where the greedy spacecraft is operating. However, the proactive behavior is best of all. This can be explained by the fact that the task map is static (no new tasks arrive in the system during the simulation) and hence the lack of external changes means that the feedforward predictions are highly accurate. As a result, in this case with no external disturbances, a more proactive or future planning approach is successful.

B. Reset Task Map

The next set of simulations keeps everything the same as before, except that the task map is altered. This new "Reset task map" has the property that at each update the ground station replaces any completed task with a new identical task. At the beginning of every update, the task map therefore always conforms to the uniform distribution. There are now fewer tasks in the system at any particular update, although the total number of tasks entering the system during the complete simulation is matched to the total number of tasks in the system at the beginning of the static task map simulation.[§§]

The reset task map is again useful for validation and understanding, but in practical terms has parallels with missions where continuous long-term measurement is of interest. For example, a climate monitoring mission might wish to repeatedly measure levels of CO_2 in the atmosphere so as to gather long-term trending information. To achieve this, every time a measurement task is completed, a new identical task is added into the system, exactly replacing (or resetting) the task that has just been completed.

Once again (Fig. 4), the obstinate behavior performs badly due to its lack of consideration. In this case, the considerate behavior is far more effective than the proactive behavior. The dynamic change in the task map after each update means that the predicted feedforward intentions are highly inaccurate. The feedforward information used by the proactive behavior is based on the fact that all of the high-priority tasks will appear to have been completed

[§§] This cannot be explicitly controlled, as the number of tasks entering the system during the 200 updates in the reset task map is dependent on the amount of duplication (or throughput) of the spacecraft.

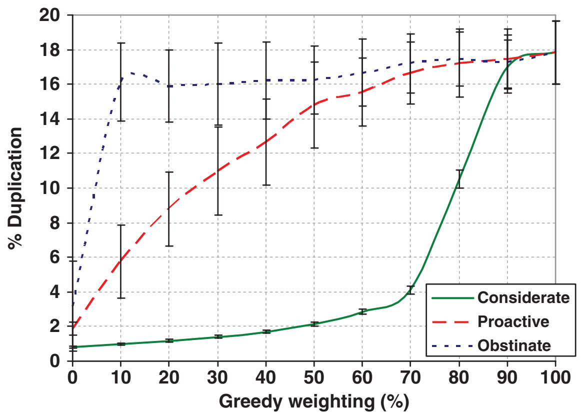

Fig. 4 Amount of task duplication generated by different behavior characteristics for a reset task map. Error bars represent 95% confidence intervals.

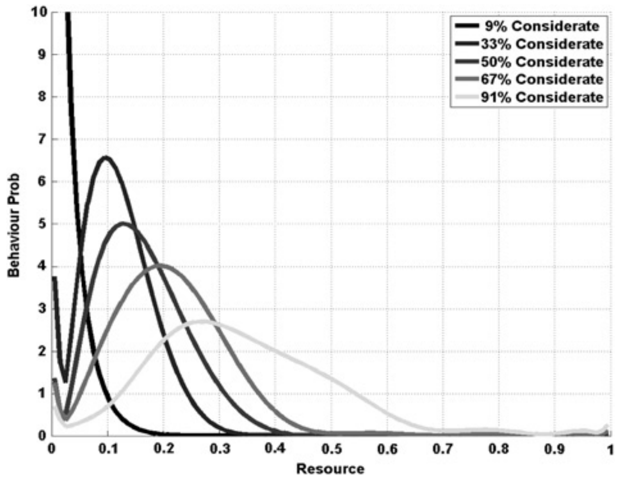

Fig. 5 A set of behaviors for the second spacecraft ranging from highly greedy to highly considerate.

each update. When replacement high priority tasks are added subsequently by the ground station, it appears that “no one is intending to do them” and hence there is no reason for the proactive behavior to avoid them. The considerate behavior, however, relies on the feedback information which is a near-perfect indicator, as the regions that were previously in contention are going to be in contention again, given the fact that the task remains unchanging (or exactly replenished). It is worth reinforcing this point by showing the actual behaviors (probability distributions) for the second spacecraft (Fig. 5). In this case, the greediest behavior (9% considerate) can be seen to be concentrating strongly on the high-priority tasks. As the amount of consideration in the behavior is increased, the behavior slowly “backs away” from the contentious high-priority regions.

C. Shifting Task Map

The previous results showed that the most difficult situation to avoid duplication was when both spacecraft had similar behaviors. To investigate this issue, the simulations are now altered so that both spacecraft have identical behavior characteristics (rather than keeping one fixed at greedy). These simulations also add more complexity to the task map by allowing it to “drift” over time. The initial distribution is uniform once again, but now completed tasks are replaced with new tasks whose resource is a random variable (also uniformly distributed). This property is introduced to model the fact that tasks generally have a time dependence or a deadline.

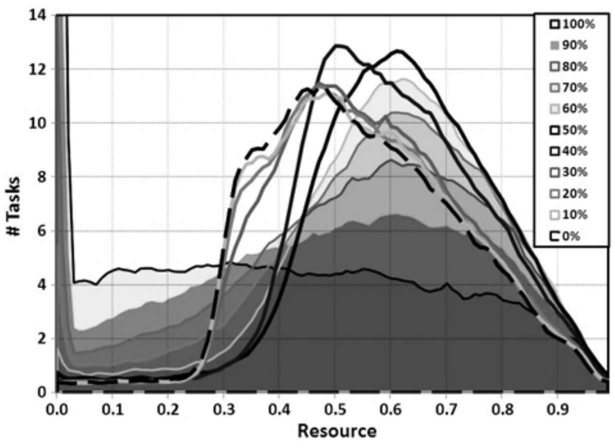

Fig. 6 Task map for various behavior combinations (in steady-state). Shading is used to distinguish curves form solid and dotted lines. Shaded curves are those with a greedy weighting ⩾50%.

The motivation behind the shifting task map is to model a situation where the resource value is actually task priority. In this model, tasks enter into the system with a particular value. As tasks remain in the system uncompleted, their priority is slowly increased, so that the entire task map appears to continuously shift toward higher priority. This queue-like mechanism ensures that if two tasks have the same initial priority, the one that has been in the system longest will have a greater chance of being performed. This is a simple model for the fact that there are a finite number of tasks that the cluster can maintain, while the ground station has a much larger buffer of tasks. As the cluster performs tasks, the ground station adds new tasks from its buffer up to the finite cluster capacity.

$$T(t,r) = T(t-1, r\cdot s) - t_c(t-1, r\cdot s) + t_a(r)\cdot \int_0^1 t_c(t-1,r)\,\mathrm{d}r \tag{10}$$

where $t_c(t,r)$ is tasks completed in update t, $t_a(r)$ is task arrival pattern, and s is amount of task shift each update.

The drift rate is such that after 100 updates a task with the lowest possible priority (1.0) will reach the highest possible priority (0.0).

Fig. 6 shows the task map in the steady-state condition with both spacecraft with identical behaviors varying the considerate/greedy configurations. Several things can be seen from this graph. When the spacecraft have a highly greedy behavior (10% considerate—the dotted line), it can be seen that the number of tasks at high priority is minimal. In fact the curve is almost cliff-like with a steep ascent at roughly 0.275, indicating how few high-priority tasks remain in the system. As the spacecraft becomes more and more considerate, the task map slowly transforms into a nearly flat distribution tailing off toward the very lowest priority. This is the considerate behavior selecting tasks well away from the area of contention (high priority) and hence there is a more even selection across the whole range resulting in a more uniform task map. There is also a tipping point as the dominating characteristic of the behavior changes from greedy to considerate (>50% considerate—the shaded curves). At this point, the shifting of the task map, combined with the lack of emphasis on high-priority tasks, means that there is a significant build up of tasks at the highest priority ($r = 0$). The rate at which tasks are being removed from the highest priority has fallen below the arrival rate of the combined task arrival and tasks shifting up toward highest priority. This bulge of high-priority task grows to dominate the task map until steady state is once again reached.

Another factor when considering the choice of behaviors is how reactive they are. Fig. 7 shows the amount of volatility in the task map (the moving average of the percentage difference between the task map profiles) in consecutive updates. Initially, there is a large percentage difference that rapidly reduces as the task map turns into steady state (in the probabilistic sense). It never reaches zero, as the system always remains dynamic. Clearly, the obstinate behavior takes the longest to converge into steady state, which is almost three times longer than the considerate behavior (taking the 4% level), with the proactive behavior in between. As expected, the considerate

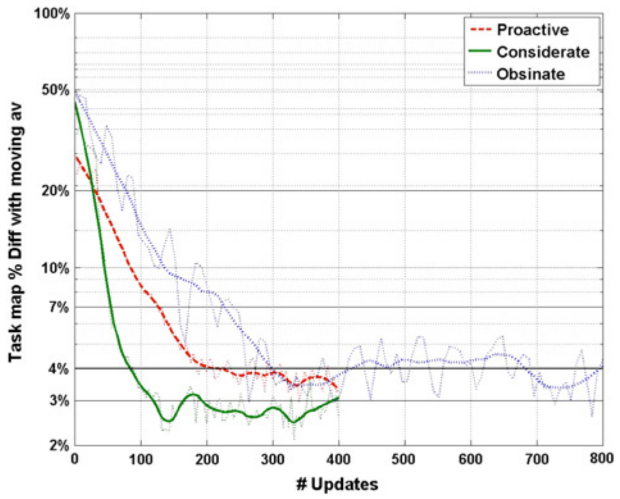

Fig. 7 The amount of time for the task map to fall into steady-state with different behaviors (non-shifting task map).

behavior is the most reactive, but the slow convergence rate of the obstinate behavior is problematic and hence, not likely to be useful in isolation. In effect, the obstinate behavior adds a lot of inertia to the simulations (deliberately so). It is therefore a strong analogy to equate obstinance to the effect of damping in spring systems¶¶ or other feedback controllers. The following sections therefore focus only on analyzing the greedy, considerate, and proactive characteristics, with obstinate left for fine tuning the damping effects.

The results shown in Fig. 8 (solid line) are obtained by varying the characteristics of the spacecraft behaviors (both identical) from 100% greedy to 100% considerate to 100% proactive and back to 100% greedy—so that they cover the full spectrum of behavior possibilities (made up of two characteristics)—forming a “triangle.” The graphs indicate that highest duplication occurs when both spacecraft are as greedy as possible. These results helped to identify the fact that it is a contention for similar tasks that is problematic rather than having identical behaviors per se. The greedy characteristic forces task selection into a particular (high priority) region whereas the proactive/considerate behavior manages to obtain a consistent duplication rate below 5% because they inherently attempt to move behaviors away from contentious regions. The phase change of the task map discussed above can also be seen as a steep gradient at around 50%, as the overall greed of the spacecraft reduces.

As the shifting task map turns into a steady state, a new performance metric of task response time can be introduced. The response time is weighted so that higher priority tasks have a bigger significance. Without this weighting, the response time is simply proportional to the throughput of the system which is already captured with the duplication measure. Response time must therefore be independent of actual task throughput, so it is normalized by the total number of tasks performed rather than the total number of unique tasks. The unit of weighted response time (WRT) is in number of tasks, with higher numbers indicating a longer time to wait. In effect, WRT can be considered as the average length of the queue for the higher priority tasks. The pure queue length is a measure of throughput which is already represented by the percentage duplication, but the weighting of WRT ensures that the response time of high-priority tasks are more significant. WRT is based on the insight that it is desirable for the higher priority tasks to be completed in preference to lower priority tasks.

$$\text{Weighted response time} = T \cdot \int_0^1 (1-r)^2 \cdot \frac{w(r)}{n(r)} \mathrm{d}r \tag{11}$$

where T is the total number of tasks performed per update, $n(r)$ the number of tasks completed with entry resource r, and $w(r)$ is $\sum_r$ number tasks waited before being performed.

¶¶ Which reduces oscillations of the system at the cost of taking longer to reach the resting state.

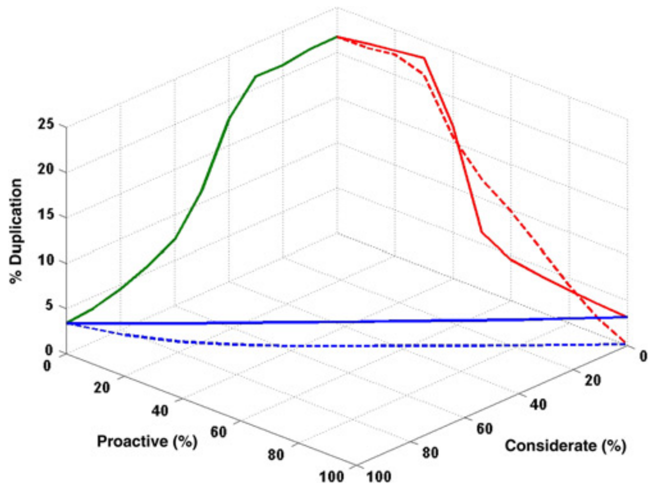

Fig. 8 Percentage duplication performance of two spacecraft with the same behavior for a range of different behavior characteristics. The solid lines represent basic behaviors with the dashed lines representing the new "self aware" behaviors that implicitly make use of the feedforward information.

It is natural to ask how the units of tasks used for WRT compare to waiting time, perhaps in number orbits. The translation is not a straightforward one as it depends on the number of spacecraft in the cluster and also the number of tasks the cluster as a whole has the opportunity to perform between updates (i.e. the communication rate). So, as an example, if an LEO orbit with an Earth imaging mission is assumed, orbits will be roughly 100 minutes with spacecraft having the opportunity to take of the order of three or four images per orbit. The controlling ground station is likely to come into range somewhere between three and five times per day giving each spacecraft the opportunity to perform of the order of 25 tasks per update (which is the value used in the previous simulations). If there are five spacecraft in the cluster a WRT value of 125 tasks represents one update, which would equate to about 5 or 6 h. Units of time clearly require assumptions about the underlying system and such orbital considerations would be inappropriate for deep space or other non-LEO missions, limiting the generality of time units.

Fig. 9 shows the response time for the three self-aware behavior ranges previously shown in Fig. 8. From Fig. 9 it is possible to see how increasingly greedy behaviors minimize the response time, as the effort is being concentrated on the "most important" tasks. Both the proactive/considerate behaviors perform tasks uniformly across the task space (on average) and hence when they operate together without any greedy weighting the response time is unchanging (and high). The slight increase in response time with behaviors that are dominated by the greedy characteristic ($>50\%$) is due to the phase change of the task map as discussed above and shown in Fig. 6.

V. Self-Awareness

At this point it is interesting to start to increase the "intelligence" of the spacecraft to see whether the performance can be improved. By intelligence in this context what is really meant is the self-awareness of the spacecraft, and how an understanding of its own actions can guide its behavior choice. Adding self-awareness is done first of all by adapting the proactive characteristic. Each spacecraft already has a knowledge about its own behavior history and its future intentions, and so it can therefore subtract its own intentions from the global intentions it receives in the feedforward information to identify what every other spacecraft is intending to do (collectively). In other words, the spacecraft is taking its own behavior into account—increased self-awareness.

It is important to note at this point that this simple introduction of self-awareness is a departure from standard swarming techniques. For example, ants have no concept of their own pheromones and do not distinguish their own markings from that of others. This self-awareness does not frequently arise in the natural world such as in ant colonies due to the large numbers. Hence, with the ants, there is a lot of redundancy as "ant power is cheap" due to the massive numbers of ants in the swarm. However, in the spacecraft case, it is very expensive to create hundreds of platforms,

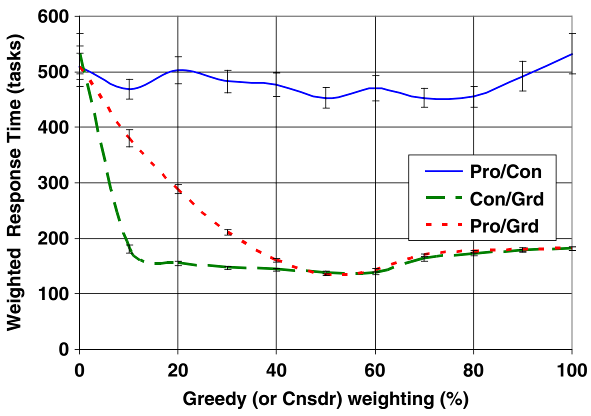

Fig. 9 WRT performance of two spacecraft with the same behavior for a range of different behavior characteristics.

and so swarm size is only a handful or tens at best. In such smaller swarms, where "spacecraft power is expensive", duplication is a far more significant problem due to the limited resource pool of the swarm. In other wards, two ants going to collect the same food is not a problem (as the wasted energy is trivial). Two rovers driving to the same location is a problem (as the wasted energy is a concern).

What is important is to use the concepts and apply those, rather than blindly copying other systems (that have different objectives and constraints). Hence, the new proactive characteristic that replaces (Eq. 22) is now defined as:

$$P_i(t) = 1 - \frac{\int_0^1 F_i(t,r)T(t,r)B_i(t,r)\,\mathrm{d}r}{[\max(F_i(t,r)) - \min(F_i(t,r))]\int_0^1 T(t,r)B_i(t,r)\,\mathrm{d}r} \tag{12}$$

where $F_i(t,r) = F(t,r) - f_i(t,r)$ and $f_i(t,r)$ is local intension of spacecraft i.

The effect of this self-aware characteristic can be seen in the dotted line of Fig. 8. Clearly, when both behaviors are 100% proactive, there is a significant improvement in the level of duplication. In fact, it reduces to almost 0% as would be expected given that with only two spacecraft in a simulation, subtracting your own intentions means that you have perfect information about the other spacecraft. A small change in the behavior characteristics has had a significant improvement. Of course, this knowledge information will no longer be perfect as the number spacecraft is increased beyond two, but it still represents a good improvement. The other thing to note is that as the behaviors shift from proactive to greedy the self-aware behavior is a smoother transition unlike the steeper change of the non-self-aware. This smooth transition is the effect of the task map shifting between two modes—swamping of high priority and keeping it in check. With the self-aware behavior, more fine-grained control is available that gradually allows the task map to change rather than a discrete jump. Of course, this self-aware behavior has no effect on behaviors that are purely greedy/considerate (as it only applies to the proactive behavior).

These simulations now move from a uniform distribution of tasks to one skewed towards higher numbers of low-priority tasks and fewer high-priority tasks which is a more realistic distribution modeling the situation where there are far more low-priority tasks than high ones. Specifically, in this case, the task arrival profile conforms to the well-known beta distribution with parameters $\alpha = 2$ and $\beta = 1$. As before, the task map continues to shift each update and a third spacecraft is now introduced, again with the same characteristics as the other two so that they all remain homogeneous. This is because much of the work on collective behaviors focuses only pairs of agents [23] as in the above simple examples. As the intention is for large numbers of spacecraft, such simplification may not be a true reflection of large scale behavior. Hence, it is important to move away from the simple cases.

Making the proactive behavior more self-aware improved the performance in the previous simulations. However, this approach could not be directly applied to either the considerate or the greedy characteristics. To add further self-awareness to these characteristics, the approach taken here is to incorporate the proactive characteristic into the other characteristics indirectly. In other words, a true greedy behavior is one that maximizes throughput as well as selecting high-priority tasks, so by reasoning about predicted future behaviors of other spacecraft, the greedy behavior now tries to choose the highest priority tasks "out of the ones that are left." This implicit behavior extends the self-awareness of the spacecraft so that they now not only consider themselves, but also make assumptions about how other spacecraft are likely to perform.

To calculate the effect of the actions of external spacecraft, it is necessary to be able to translate the probabilities (of behavior and the feedforward intentions) into expected numbers of tasks to be performed. For implementation, the continuous distributions are already discreetized into 100 bins each containing a certain number of tasks. Hence, given a certain number of tasks in a particular bin, the probability of selecting from that bin (determined by the behavior probability) and the total number of tasks to be performed in the update—what is the expected number of tasks that will be performed from that particular bin? The problem is complicated by the fact that tasks are selected sequentially (i.e. without replacement) affecting subsequent selections. Moreover, the behavior means that the probability of selection from a particular bin is weighted. Altogether therefore these constraints point to the use of the Wallenius univariate non-central hypergeometric distribution [25] to calculate the expected number of tasks selected from the bin.[##]

$$1 = \frac{\mu}{t_b} + \left(1 - \frac{n - \mu}{t_r}\right)^{p_b/1-p_r} \tag{13}$$

where μ is expected no. of tasks performed, t_b no. of tasks in bin b, t_r no. of other tasks, n no. of tasks performed per update and p_b is behavior probability.

Unfortunately, there is no closed form analytical solution to calculate the expected mean, but it can be numerically approximated by the solution to Eq. (13). Numerical approximation using Taylor expansion is clearly a computationally expensive process especially when this function is embedded within the fitness function of the on-board genetic algorithm search. However, in principle, this issue can be sidestepped by calculating the results off-line and using a lookup table on-board the spacecraft. This lookup function can be used to calculate the expected number of tasks that will be performed by the local behavior, and also by using the feedforward intentions information, the expected number of tasks performed by the external spacecraft.[***]

The next step is to calculate the expected overlap (or task duplication). The number of tasks in the bin is known, as are the expected number of selections from the bin by the local and external spacecraft, calculated using Eq. (19). The expected intersection can therefore be derived by using the (centralized) hypergeometric distribution[†††] [25]. The straightforward formula for calculation of expected intersection can be wrapped to give a function that expresses the percentage of unique tasks that are expected to be performed.

$$I(t, b_\mu, e_\mu) = \frac{b_\mu - (b_\mu \cdot e_\mu/\mathrm{t})}{b_\mu} \tag{14}$$

where t is no. of tasks in bin, b_μ expected no. of tasks performed locally by Eq. (19), and e_μ is expected no. of tasks performed externally by Eq. (19).

This function can now be incorporated as a "modifier" into the above fitness function equations (Eqs 20, 21, 22, 23 and 28) inside the integrals. This modifier effectively reduces the probability so as to only consider unique tasks performed by this particular behavior, in the probabilistic sense. This can be done with relatively low overhead using lookup tables as mentioned above and also noting that the performed tasks by the external spacecraft is invariant in between updates, and so for practical implementation can be optimized as a single calculation despite the repeated use of the calculation.

[##] The standard analogy for this distribution is of a bag containing a finite number of both red and black balls. There is a propensity to select red balls with some probability. If a certain number of draws are made from the bag, without replacement, what is the expected number of red balls that will be drawn?
[***] This, of course, is not strictly accurate as the external spacecraft may well overlap with each other, however, it is considered to be a reasonable approximation given that other approximations are also present.
[†††] This is a simpler hypergeometric distribution, so in the balls analogy, the probability of selecting red and black balls is identical. For example, this is used in standard lottery mathematics, to calculate the probability of intersection of a player's numbers and the machine drawn numbers.

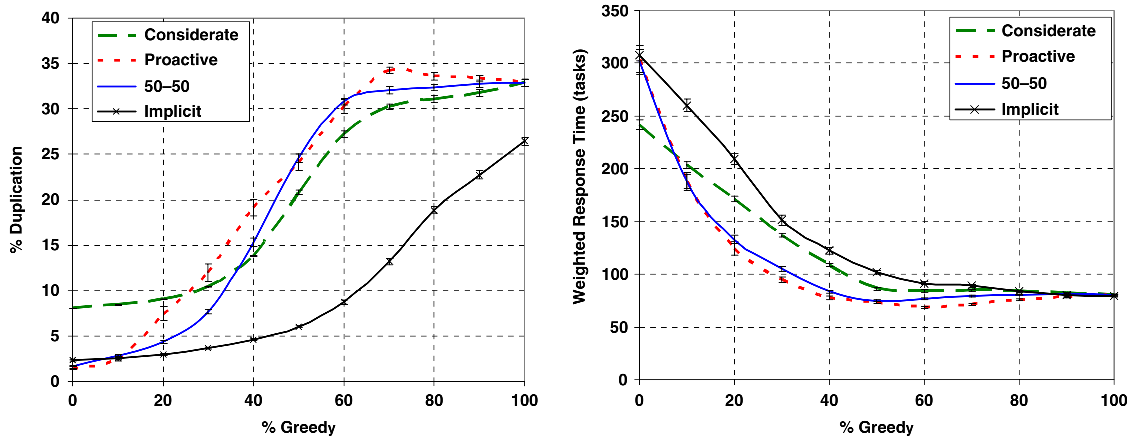

Fig. 10 Percentage duplication and a WRT for three homogeneous spacecraft for considerate and proactive ranging to greedy as before. Also shown is a behavior which has equal amounts of considerate and proactive (remainder greedy) called 50–50, and the new implicit behavior that ranges from greedy to considerate.

Fig. 10 shows a comparison of the new implicit behavior (ranging from 100% greedy to 100% considerate) with the other explicit behaviors that have been used up to this point. There is a huge improvement in the levels of task duplication (throughput) that can be achieved—even when all three spacecraft are 100% greedy. It must be noted that the duplication levels are slightly higher than the previous results because there are now three spacecraft rather than two making the problem more challenging. The graphs also include a comparison with an additional new behavior that includes all three (greedy, considerate, and proactive) components explicitly, which again has lower throughput performance. These results are similar for other weightings of the characteristics, showing that the implicit behavior is vastly superior. What is important is that the implicit behavior performs better than any linear combination of the characteristics with explicit proactiveness.

Implicit behavior does have a slightly higher WRT overall. However, this difference becomes negligible to non-existent in cases where the spacecraft are highly greedy, i.e. when WRT is of most interest. The implicit behavior loses out only in situations where response time is not so critical but where maximizing throughput is. The implicit behavior can therefore be considered to be more flexible over a larger range of performance values.

From this point on, all simulations are based on behaviors using the implicit definition.

VI. Performance Trade-Offs

As has been alluded to previously, the stigmergy system has some fundamental performance trade-offs. Namely that it is possible to minimize duplication, or to minimize the response time of high-priority tasks, but not to achieve both simultaneously. This is a true trade-off, because which end of this performance spectrum is more desirable to the spacecraft operators is not obvious. In some situations, achieving a good background throughput would be desirable, but in others, responding rapidly to tasks of interest may well be more important. It is likely however that in the normal case, operators will want a little bit of both. To understand the reason that this trade-off actually exists, it is necessary to look in detail at the underlying system state.

Fig. 11 shows the results of five (an increase from the previous simulations) homogeneous spacecraft[‡‡‡] operating on the shifting task map. Again the two extremes of all 100% greedy and all 100% considerate spacecraft are considered. The graphs show the resulting steady-state task maps, together with the global environment information. The left-hand graph shows all five spacecraft with 100% considerate behaviors (including implicit proactiveness). The task map conforms to the roughly skewed distribution of the task arrival pattern, with a slight bulge due to the

[‡‡‡] In other words, they all have the same behaviors and capabilities at the start of the simulation.

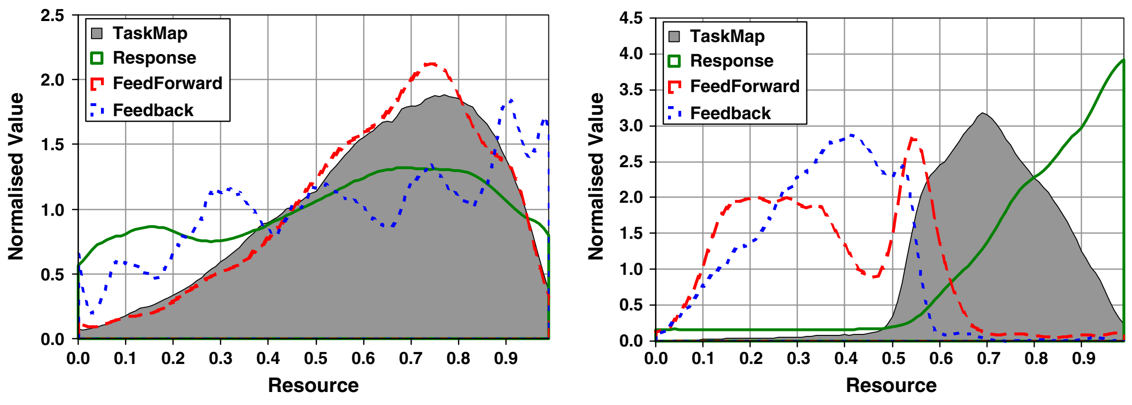

Fig. 11 Task maps and environment information in the steady-state condition for the implicit 100% considerate behavior (left) and the implicit 100% greedy behavior (right).

shifting property of the tasks. The feedforward (intentions) information shows how the behaviors are concentrated almost exactly in line with the task arrival pattern and the feedback (duplication history) is distributed across the spectrum. Similarly, response time for tasks of different resource levels varies only slightly. In stark contrast, the 100% greedy behaviors (right-hand side) concentrate entirely on the high-priority region to the extent that there are almost no tasks left. Feedforward intentions and feedback duplication history both show how all the tasks are being performed in that region with a corresponding build up of low-priority tasks. The response time curve also indicates how response time is minimal for high-priority tasks and then rapidly escalates for the lower priority tasks exactly as is desired by the behavior characteristics.

This is mirrored in the actual behavior choice probabilities as shown in Fig. 12. Concentrating on higher priority tasks (right-hand side) forces all the spacecraft to operate in a smaller, overlapping region with the behaviors effectively surfing the crest of the task standing wave. The nature of the behaviors, being greedy and not considerate, means that the tasks selections of the spacecraft are naturally contending with each other, others making the problem of avoiding behavioral overlap far more challenging. A mitigating factor, however, is the fact that although total duplication is increasing, it is the higher priority tasks that are being duplicated. This is likely to be less detrimental to performance (and could even be considered to add redundancy) compared to duplication of low-priority tasks, which could be considered more "wasteful." The left-hand graph shows how the 100% considerate behaviors in contrast are managing to separate well to avoid overlap. However, the trade-off is that they are less focused on the high-priority region and hence the response time suffers.

Assigning all spacecraft the same behavior characteristics (so they are homogeneous) allows for some good insight into the problem, however, in practice, it is likely that spacecraft behaviors will vary across the cluster so that they are heterogeneous.§§§ The goal is ultimately to have spacecraft not only with different behaviors, but also different payloads and performance characteristics. Fig. 13 shows the results of repeating these simulations on the shifting task map with five spacecraft, but rather than giving all spacecraft the same behavior, the greedy/considerate ratio is adjusted to give more heterogeneous behaviors.¶¶¶ Hence, although the average greedy (or considerate) weighting for the five spacecraft remains constant, the standard deviation of the greedy (or considerate) weighting across the cluster is varied.###

Fig. 13 reinforces the point that, as spacecraft become more heterogeneous (increasing standard deviation), duplication is reduced. Similarly, this spreading out of the behaviors tends to focus attention away from the higher priority task and so WRT increases. This is exactly the same effect as was seen in the homogeneous case. As spacecraft

§§§ Although physically they have the same capabilities, only the behaviors are changed.
¶¶¶ That is, the behaviors are linearly spread in terms of behavior characteristics. So, for example, rather than being all homogenous with 80% greedy (20% considerate), they will be spread from 100% greedy (0% considerate) to 60% greedy (40% considerate).
That is, homogeneous spacecraft have a standard deviation of zero as they are all the same.

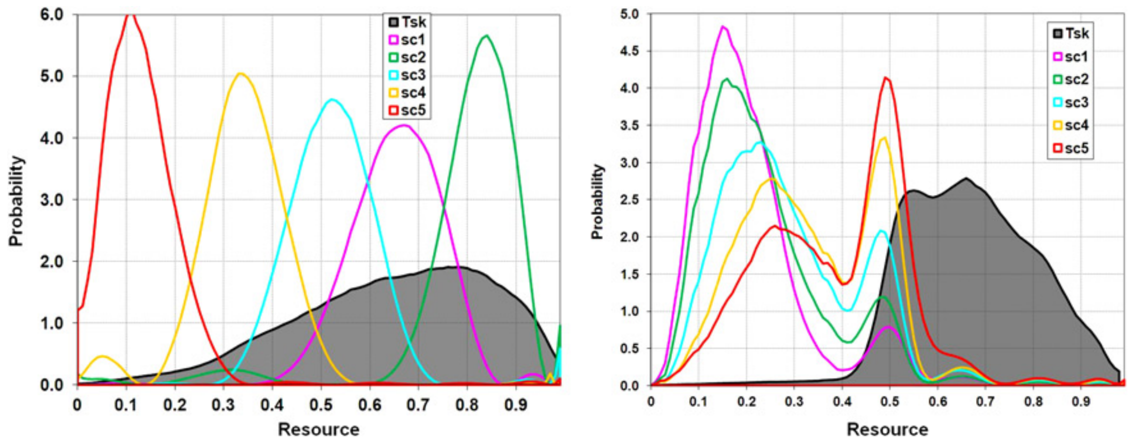

Fig. 12 The behaviors of five homogeneous spacecraft (and global task map) in steady-state with 100% considerate (left) and 100% greedy behaviors (right).

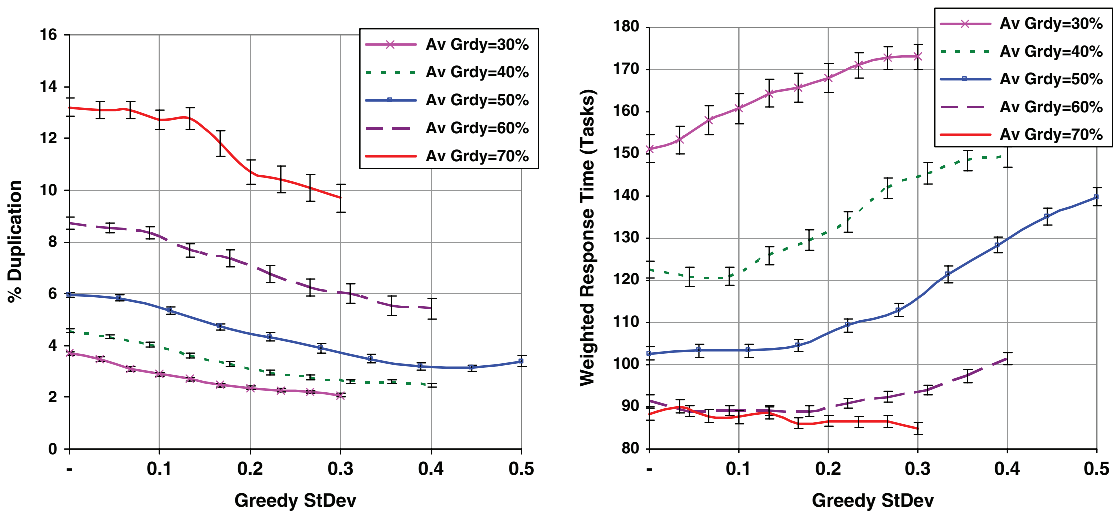

Fig. 13 The effect of keeping the sum of the behavior characteristics in a cluster constant and varying the standard deviation of these characteristics between the spacecraft. Each line represents a different sum total.

behaviors become more heterogeneous, they are inherently competing for different regions of the task map and so the amount of task contention is lower. In other words, task duplication again decreases, although the response time suffers.

The second important thing to note is that the dominating factor is the overall mean of the behavior characteristics with the difference between the spacecraft as a lower order effect. Hence, it is a reasonable simplification to consider only the overall mean behavior characteristics of the cluster if the standard deviations are kept to moderate values.

VII. Power Modeling

Having considered the implementation of the algorithms on an individual spacecraft, this case study now returns to the performance of multiple spacecraft. In this case, the task map is adjusted to conform to a more realistic spacecraft scenario in which the resource considered is the amount of power required for each of the tasks.

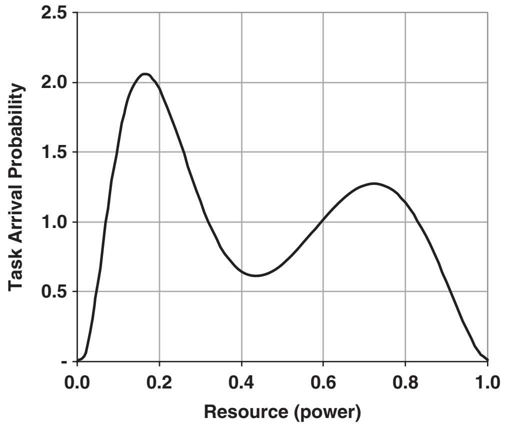

Fig. 14 The arrival distribution of tasks with varying power requirements. For reference this curve is mathematically defined as (β(6,3) + β(3,12))/2.

This “double hump” arrival distribution (Fig. 14) is chosen to model a spacecraft with two types of instrument on-board. Each instrument uses a different average amount of power, although there is still some specific variability such as the size of image, or for how long to measure. Equal numbers of tasks are assigned to each of the instruments. For the lower power case, the standard deviation is set to be smaller, resulting in a higher narrow distribution of task power requirements. In such a scenario, the power requirements of individual tasks remain fixed, irrespective of how long they remain in the system, and so the shifting model (that was previously used when considering priority in Section IV) is not used.

Another property of this scenario is that there is only a finite amount of power available on-board spacecraft per update (due to the charging effects of the solar panels). To model this, rather than having a fixed task completion rate, as before, the number of tasks performed by the spacecraft is proportional to their associated power requirements. The spacecraft can do a larger number of low-power tasks or a smaller number of high-power tasks, up to some maximum power threshold. For consistency with previous simulations, this threshold is chosen to match the fixed completion rate on average. With the power required per task dimension ranging between 0 and 1, the finite amount of power available for each update is set to be 12.5 (so that on average 25 tasks per update will be completed as was generally the case previously).

Tasks still inherently have some time dependence and hence it is unlikely that a spacecraft will be able to do hundreds of very low power tasks in a single update. This time dependence is not explicitly considered in this system as it is devolved to the lower level scheduling operations agent (NEAT architecture). For simulation purposes, the maximum number of tasks that can be completed per update is a reasonable approximation and this is added as another constraint. Spacecraft can now therefore complete a maximum of 25 tasks and use a maximum of 12.5 power per update.

A simulation of five spacecraft all with heterogeneous behaviors is now considered. The behavior characteristic weightings are spread so that at one extreme there is a 100% greedy spacecraft. Three different cluster configurations are considered so that the other extreme is 100% proactive in the first case, 100% considerate in second, and in a third case it is 50% considerate and 50% proactive. Spacecraft in between the two extremes are linearly spaced.

Fig. 15 shows the results of the amount of power used per update for the five spacecraft (on the left) and a number of tasks performed per update on the right. It is clear that the greediest spacecraft aggressively try to perform tasks requiring the least amount of power, and are constrained by the throughput limitation of the maximum number of tasks per update. The greediest spacecraft therefore complete the maximum number of tasks, but use very little power. In contrast, the non-greedy spacecraft are pushed into performing tasks that have much higher power requirements and are constrained by the availability of power. The spacecraft therefore use their maximum available requirement, but complete fewer tasks. The three data series show similar performance patterns. This is because although they

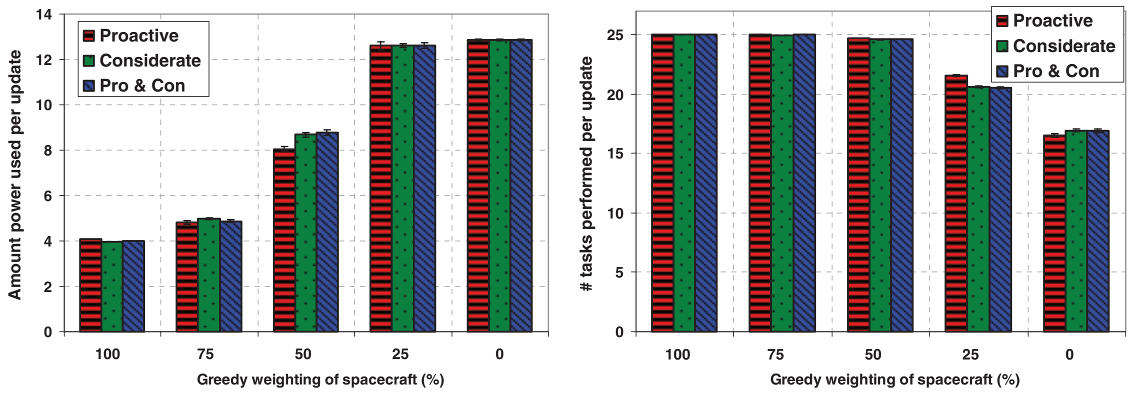

Fig. 15 A cluster of five spacecraft ranging from 100% greedy, in a linear spread, to 100% considerate/proactive or considerate and proactive 50–50. The graphs show how greedy spacecraft are throughput limited, whereas the other extreme are power limited.

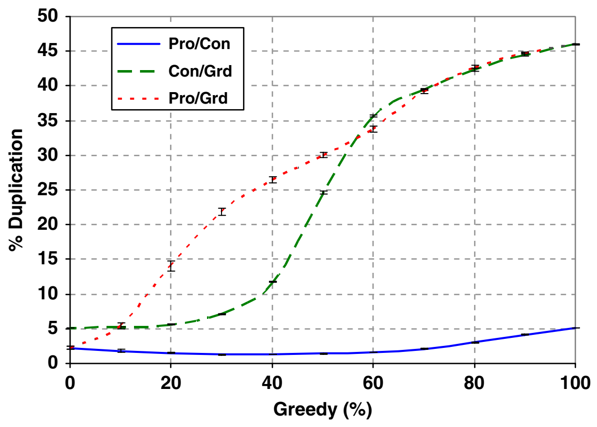

Fig. 16 Performance of three identical spacecraft on the power double hump problem.

have different behavior spreads, the fact that they are all self-aware (with implicit proactivity) there is very little difference between the performance of the proactive and considerate behaviors.

Despite this extra power constraint, the performance of the system remains in line with previous results. This can be seen in Fig. 16 which repeats the performance metrics used previously (for example, Fig. 8). In this case, three spacecraft all have identical behaviors ranging across different characteristic values. As expected, when the spacecraft are all very greedy, the amount of task duplication is highest. Of course, in this context where the resource represents power rather than priority, the response time is meaningless.

In this setup, where power is really of interest, what is important is the amount of power each spacecraft is using. This was not problematic when the task resources were considered as priority, but when it is a physical property this introduces the potential for individual spacecraft to become overloaded whilst others are under loaded. In other words, each spacecraft will have a higher propensity to perform different types of task, which, over the long term, will lead to an imbalance of resource usage. It is therefore of significant importance to allow some kind of load balancing so that the overall lifetime of the cluster can be extended without burning out one of the spacecraft, by depleting all its resources.

This load balancing can naturally be achieved by dynamically varying the behaviors of the spacecraft over time to load balance the cluster, so that all spacecraft can take turns in the most resource intensive regions. This behavior

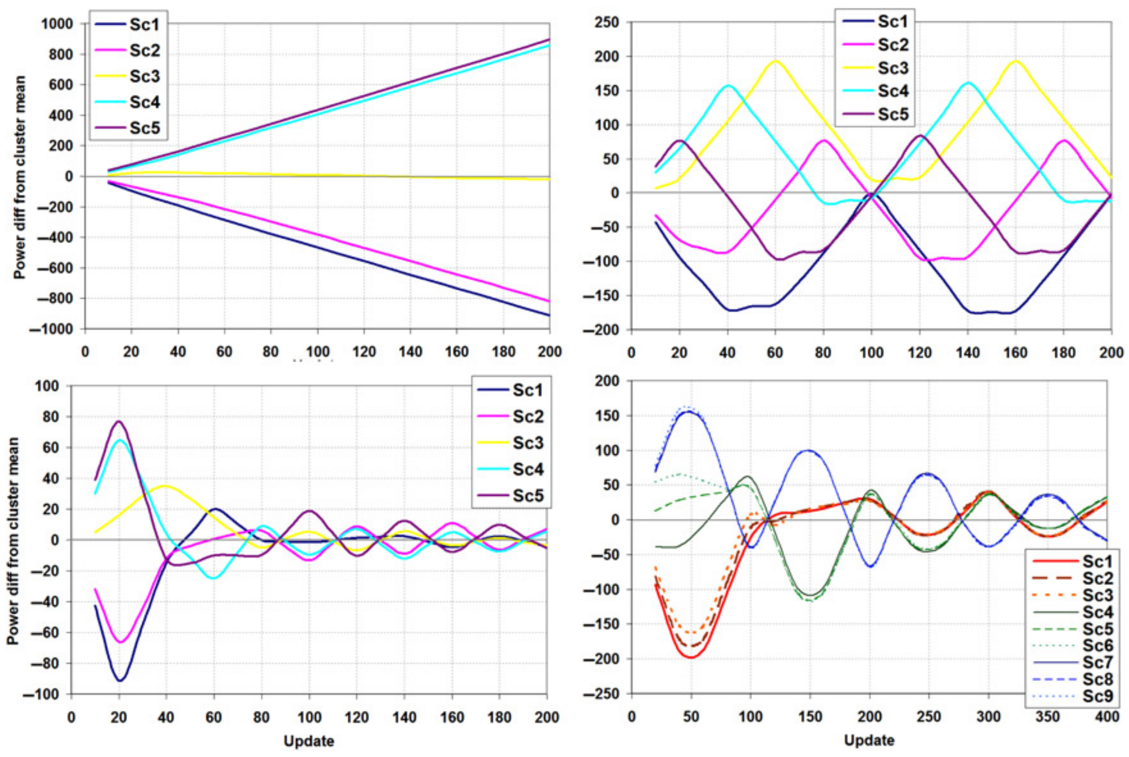

Fig. 17 The difference in the amount of resource usage from the cluster average without supervisor adjustment (top left), with round-robin adjustment (top right) with feedback adjustment (bottom left), and with feedback hierarchical adjustment (bottom right).

adjustment takes place at a higher system level than the individual spacecraft. In effect, this is introducing the concept of a hierarchical control to the simulations. The supervisor or ground station is responsible for coordinating its subordinates so as to load balance between them and ensure even resource usage. For this example, three behavior adjustment policies are considered (listed in order of increasing sophistication):

- No adjustment.
- Round robin. The spacecraft behaviors are cyclically altered.
- Feedback. The spacecraft are ordered based on the amount of resource they have used and then behaviors assigned so as to reduce the imbalance.

As before, the simulations have five spacecraft with linearly spaced behaviors from 100% greedy to 100% considerate. The graphs of Fig. 17 show the difference in amount of power used by each of the five spacecraft for the different adjustment strategies. Behavior adjustment takes place every 20 updates (i.e. roughly once a month for an LEO consolation).

The results are reasonably intuitive. No adjustment (top left) causes the amount of power usage to continually diverge. The round robin approach (top right) keeps the difference in resource usage within bounded ranges and displays a cyclic pattern. The problem with round-robin is that if there is a particular burst of tasks or other significant event, a particular spacecraft may well be "unlucky" and be forced to consume a lot of power. The round-robin approach means that this offset can never be undone. Clearly, the feedback responsive approach (bottom left) is the most suitable, almost negating the resource usage over time. This is because behaviors are matched to the current status of the system, so that if a spacecraft is particularly "unlucky" it will subsequently be compensated. When examining the three graphs, it is important to note the magnitude of difference in power difference (i.e. the scale of the y-axis).

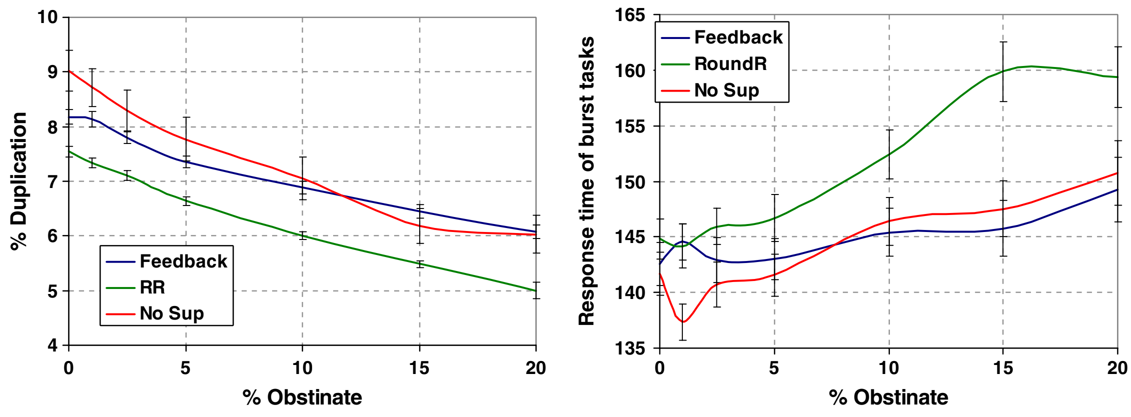

Fig. 18 Effect of adding the obstinate characteristics to spacecraft behaviors to reduce duplication during load balancing (left) but this increases the response time for a burst of high-priority tasks (right).

This feedback load balancing can also be extended to a hierarchical system. Fig. 17 (bottom right) shows nine spacecraft organized into three teams of three.**** In this case, one of the three spacecraft in a particular subteam is designated as supervisor and adjusts the behaviors of the members of its subteam using the feedback adjustment strategy. The supervisor aggregates the tasks performed by the members of the subteam before passing them down to the ground station, so that from the ground station's perspective there are only three super-spacecraft to manage. The ground station now balances these three super-spacecraft in the same hierarchical manner, although it does so at a lower frequency—every 50 updates.

Unfortunately, load balancing the spacecraft does not come with zero cost. As the proactive characteristic uses future intentions, if the behaviors instantaneously change, this estimate becomes quite inaccurate as the system is forced to "relearn" the new state. This has the effect of increasing the amount of task duplication or increasing the response time. One simple solution is to reintroduce the concept of the obstinate behavior, in order to alleviate this effect. The obstinate behavior effectively introduces inertia into the system. By making each of the spacecraft slightly obstinate, when the other characteristics are changed, the new behavior will slowly evolve rather than instantaneously switch. This can be seen in Fig. 18 (left) where increasing the amount of obstinate characteristic in the behaviors reduces the amount of duplication.

Once again, there is another trade-off to the obstinate behavior. Although the inertia effect is good at minimizing duplication of the behavior switch, it does impact on the responsiveness of the behavior to a dynamic task map. Hence, for a task map that has a burst of high priority tasks, obstinate behaviors are far less likely to instantaneously switch over to perform the tasks, as can be seen in Fig. 18 (right). A more successful strategy might be to vary the degree of obstinance over time so that it is increased during the load balancing exercise although can be dynamically dropped if a burst of high-priority tasks enters the system.

VIII. Discussion

With the ground station supervisor adjustments, the indicator used was the amount of resource consumed by each of the spacecraft. However, this could equally well be transformed into amount of resource (e.g. propellant) remaining. In such a situation, this small change can be used to adjust the spacecraft in order to maintain resource levels and extend the lifetime of spacecraft. Of course, such decisions can be left to the operators, but in general it must be assumed that it is more desirable for all spacecraft be operational for as long as possible rather than

**** Nine spacecraft were chosen so as to have a natural hierarchical division of three per team. Five spacecraft were chosen in the other instances to avoid over complicating the graphs with nine spacecraft. Whatever the number of spacecraft, the resultant behavior is similar, with the actual numbers are representative chosen here for clarity an explanation. This is also revisited in the discussion below.

"burning out" individual spacecraft well in advance of the remaining spacecraft. This system allows the spacecraft operators to move away from micromanaging the specific task operations and toward subtle adjustments of behavior goals which both reduces the problem complexity for the operators and communications requirements of the system.

This article has started to demonstrate the feasibility of using stigmergy for multiple spacecraft coordination. In such a system there are a large number of parameters to configure many of which can have significant impacts on the overall dynamics of the system. A lot more research is needed into this topic to vary the parameters not adjusted here, for example evaporation rates, communication rates, task error rates, etc. What is clear however is that for a given situation set of parameters can always be found that achieves highly desirable performance. However, as this system displays emergent properties, general rules for being able to methodically find the optimal parameter sets are not possible to identify [26].

One of the key benefits to this proposed system is the minimal networking infrastructure required. This is crucial as swarm and cluster missions are primarily envisaged for small platforms with limited capabilities. Tightly constrained mass and power budgets mean that additional antenna (and possibly pointing) hardware is difficult to incorporate, hence a coordination system that does not require intersatellite links is a huge benefit. This article has demonstrated coordination of five spacecraft without intersatellite links; however, previous work has shown up to 18 in a single team [21]. The hierarchical supervisor architecture using nine spacecraft begins to include intersatellite links but only for a subset of the spacecraft. Such an architecture is again likely when considering deployment scenarios such as a group of picosatellites and a single nanosatellite acting as a ground station relay [27]. As any intersatellite links introduced to the system raises the possibility of enhancements by the use of rumor protocols [28] (or other mechanisms) that efficiently synchronize the environment information between spacecraft. Such synchronization that would be achievable in a hybrid system effectively means increasing the rate at which environment updates arrive which can only improve system performance.

IX. Conclusion

This article has looked at the behavioral characteristics necessary to give rise to multiple spacecraft collaboration without using intersatellite links. Collaboration is achieved by using the principles of stigmergy which allow spacecraft to coordinate indirectly through a common environment that is periodically broadcast from the ground station. Such a communication architecture is inherently scalable and suitable for implementation on small spacecraft. Hybrid architectures have also been investigated where some of the spacecraft do have intersatellite links capabilities and can act as "supervisors" effectively partitioning the cluster into teams. This hierarchical arrangement is another key tenant of making the system scalable to potentially hundreds of spacecraft.

Investigation of the behaviors has revealed that by making them more self-aware or "more intelligent" superior performance can be achieved. Fundamental trade-offs do exist however such as between maximizing throughput and maximizing the response time of high-priority tasks.

The crucial benefit of the system is that it has the ability to free the ground station operators from the micromanagement of individual spacecraft, something that will certainly be needed for the more distributed and aspirational swarm missions of the future.

Acknowledgments

This work was entirely funded by the Autonomy group of the Surrey Space Centre, part of the University of Surrey, Guildford, UK.

References

[1] Clement, B., and Barrett, A., "Coordination Challenges for Autonomous Spacecraft," in *An Application Science for Multi-Agent Systems*, edited by T. A. Wagner, Vol. 10, Springer, USA, 2004, pp. 7–26. http://dx.doi.org/10.1007/1-4020-7868-4_2.

[2] Hinchey, M. G., Rash, J. L., and Truszkowski, W. F., "Autonomous and Autonomic Swarms," *Proceedings of the International Conference on Software Engineering Research and Practice*, 2005, Las Vegas, NV.

[3] Brown, O., and Eremenko, P., "Fractionated Space Architectures: A Vision for Responsive Space," *Proceedings of the 4th Responsive Space Conference*, 2006, Los Angeles, CA.

[4] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Lee, R., Mandl, D., Frye, S., Trout, B., Hengemihle, J., D'Agostino, J., Shulman, S., Ungar, S., Brakke, T., Boyer, D., Van Gaasbeck, J., Greeley, R., Doggett, T., Baker, V., Dohm, J., and Ip, F., "The EO-1 Autonomous Science Agent," *Proceedings of Autonomous Agents and Multi-Agent Systems*, 2004, New York City, USA.

[5] Muscettola, N., Fry, C., Rajan, K., Smith, B., Chien, S., Rabideau, G., and Yan, D., "On-board Planning for New Millennium Deep Space One Autonomy," *Proceedings of the IEEE Aerospace Conference*, 1997, Snowmass, CO.

[6] Carrel, A., "Adaptive Evolutionary Decision Making for Autonomous Spacecraft," PhD Thesis, Surrey Space Centre, University of Surrey, 2007.

[7] Carnelli, I., Dachwald, B., and Vasile, M., "Evolutionary Neurocontrol: A Novel Method for Low-Thrust Gravity-Assist Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 2, 2009, pp. 616–625. http://dx.doi.org/10.2514/1.32633

[8] Schetter, T. P., Campbell, M. E., and Surka, D. M., "Comparison of Multiple Agent-based Organizations for Satellite Constellations (TechSat21)," *Proceedings of the 13th International Florida Artificial Intelligence Research Society*, 2000, Orlando, FL.

[9] Zetocha, P., "Satellite Cluster Command and Control," *Proceedings of the IEEE Aerospace Conference*, 2000, Big Sky, MT, USA.

[10] Richards, R. A., Houlette, R. T., and Mohammed, J. L., "Distributed Satellite Constellation Planning and Scheduling," *Proceedings of the 14th International Artificial Intelligence Research Society Conference*, 2001, FL.

[11] Clement, B., Barrett, A., and Schaffer, S., "Argumentation for Coordinating Shared Activities," *Proceedings of the International Workshop on Planning and Scheduling for Space*, 2004, Darmstadt, Germany.

[12] Bonnet, G., and Tessier, C., "Collaboration among a Satellite Swarm," *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007, Honolulu, HI.

[13] Gao, S., Brenchley, M., Unwin, M., Underwood, C. I., Clark, K., Maynard, K., Boland, L., and Sweeting, M. N., "Antennas for Small Satellites," *Proceedings of the Antennas and Propagation*, 2008, Loughborough, UK.

[14] Hinchey, M. G., Rash, J. L., Truszkowski, W. F., Rouff, C. A., and Sterritt, R., "Challenges of Developing New Classes of NASA Self-Managing Missions," *Proceedings of the 11th International Conference on Parallel and Distributed Systems*, 2005, Fukuoka, Japan.

[15] Curtis, S. A., et al., "Use of Swarm Intelligence in Spacecraft Constellations for the Resource Exploration of the Asteroid Belt," *Proceedings of the 3rd International Workshop on Satellite Constellations and Formation Flying*, 2003, Pisa, Italy.

[16] Grasse, P. P., 'The Automatic Regulations of Collective Behavior of Social Insect and 'Stigmergy'," *Psychologie normale et pathologique*, Vol. 57, 1960, pp. 1–10.

[17] Karuna, H., Valckenaers, P., Constantin, Z., Van Brussel, H., Bart, S. G., Hoelvoet, T., and Steegmans, E., *Self-Organising in Multi-agent Coordination and Control Using Stigmergy*, Lecture Notes in Computer Science, 2004, Springer, Berlin, Heidelberg.

[18] White, T., *Expert Assessment of Stigmergy: A Report for the Department of National Defence*, 2005, School of Computer Science, Carleton University, Canada.

[19] Bonabeau, E., Dorigo, M., and Theraulaz, M., *Swarm Intelligence: From Natural to Artificial Systems*, 1999, Oxford University Press Inc., USA.

[20] Parunak, H. V. D., Purcell, M., and Connell, R. O., "Digital Pheromones for Autonomous Coordination of Swarming UAV's," *Proceedings of the 1st AIAA Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, and Operations*, 2002, Norfolk, VA.

[21] Tripp, H., and Palmer, P., "Distributed Behavioural Coordination for Satellite Clusters," *Proceedings of the International Astronautical Congress 2008*, 2008, Glasgow, UK.

[22] Tripp, H., and Palmer, P., "Distribution Replacement for Improved Genetic Algorithm Performance on a Dynamic Spacecraft Autonomy Problem," *Journal of Engineering Optimization*, Vol. 42, No. 5, May 2010, pp. 403–430.

[23] Panait, L., and Luke, S., "Cooperative Multi-Agent Learning: The State of the Art," *Autonomous Agents and Multi-Agent Systems*, Vol 11, No. 3, 2005, pp. 387–434. http://dx.doi.org/10.1007/s10458-005-2631-2

[24] Chen, W., "Performance Modelling of Imaging Service from Earth Observation Satellites," PhD Thesis, Surrey Space Centre, University of Surrey, 2007.

[25] Fog, A., "Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution," *Communications in Statistics, Simulation and Computation*, Vol 37, No. 2, 2008, pp. 258–273.

[26] Dessalles, J. L. and Phan, D., "Emergence in Multi-Agent Systems: Cognitive Hierarchy, Detection, and Complexity Reduction Part I: Methodological Issues," *Proceedings of the Symposium in Agent-based Computational Methods in Finance, Game Theory and Their Applications*, 2005, Lille, France.

[27] Baker, A. M., Bridges, C. P., Underwood, C. I., Vladimirova, T., Alex da Silva, C., and Barnhart, D. J., "Thinking Outside the Cube: A Radical New Approach to Nanosatellite Missions," *Proceedings of the International Astronautical Congress*, 2008, Glasgow, UK.
[28] Datta, A., Quarteroni, S., and Aberer, K., "Autonomous Gossiping: A Self-organizing Epidemic Algorithm for Selective Information Dissemination in Mobile Ad-hoc Networks," *Proceedings of the International Conference on Semantics of a Networked World*, 2004, Paris, France.

Ella Atkins
Associate Editor