Self-Organizing MPS for Dynamic EO Constellation Scenarios

C. Iacopino * and **P. Palmer** †
Surrey Space Centre,
University of Surrey,
Guildford, GU2 7XH, United Kingdom

N. Policella ‡ and **A. Donati** §
ESOC, European Space Agency,
Robert-Bosch-Strasse 5,
Darmstadt, 64293 Germany

A. Brewer ¶
Surrey Satellite Technology Ltd.,
Surrey Research Park,
Guildford, GU2 7YE, United Kingdom

Abstract

Earth Observation (EO) constellations have the potential to offer critical services for the society such as global monitoring and disaster management. A first example of this increasing interest is the Global Monitoring for Environment and Security (GMES) program. Multiple platforms however, open new challenges in terms of coordination and high responsiveness, mainly in critical scenarios.

This paper presents a new concept for ground-based automated planning & scheduling systems applied to distributed platforms. The system developed is a self-organizing multi agent system inspired by ant colonies; it is able to find solutions that maximize the satellites' efficiency and minimize the duplications among the satellites' plans. Moreover, this approach offers high-level of adaptability and responsiveness. The EO constellation Disaster Monitoring Constellation (DMC) is used as case study as it represents a dynamic distributed problem. An empirical evaluation presents the algorithm capabilities. This approach aims at extending automated mission planning applications to real constellation scenarios.

Multiple platforms such as constellation, cluster or swarm of satellites are the new trend of the space missions as they offer a number of benefits over single monolithic spacecraft and are already largely adopted for communication, geo-location (GPS) and meteorology purposes. However this paradigm introduce new elements of complexity, which hinder their application to a wider number of space missions such as Earth Observation. One of these challenges concerns the mission planning because the coordination between the spacecraft is one of the critical aspects for a distributed mission. Techniques from the field of Automated Planning & Scheduling can help to handle this complexity and enable new operational concepts. A number of missions have already demonstrated the benefits of these technologies for Operations but so far mainly for single platforms and those solutions are not necessarily transplantable in distributed contexts.

Focusing on the Earth Observation (EO) constellations scenario, one of few autonomous Operations examples that have been demonstrated in space, is the tandem mission TerraSAR/TanDEM-X (Lenzen et al. 2011), where basic functionalities of automated scheduling have been implemented though without optimizing the resources. The real big challenge is coordination and optimization at the same time. A number of studies have recently shown interest for the disaster management, focusing on sensorweb (Chien et al. 2011; Mandl et al. 2008; Chien et al. 2005) or just on Earth Observation constellations (De Florio 2006; Pralet, Verfaillie, and Olive 2011; Grasset-Bourdel, Verfaillie, and Flipo 2011; Raghava Murthy et al. 2010; Wang and Tan 2008). Most of them tried to reduce the coordination aspect to an optimization problem and to solve it with classic techniques such as greedy (Wang and Tan 2008; Pralet, Verfaillie, and Olive 2011), backtracking (Grasset-Bourdel, Verfaillie, and Flipo 2011) or simple heuristics (De Florio 2006). In these cases either they did not achieve efficient solutions either they considered small problems (reduced number of spacecraft). Moreover, a big limitation of these works is not considering the dynamics of the problem itself. This scenario needs to be faced as a dynamic environment. In case of the GMES or of the Charter system, the five ESA spacecraft devoted to Earth Observation (Sentinel-1 to Sentinel-5) need to cooperate with other existing and/or planned missions provided by ESA, EUMETSAT, other national agencies or private companies such as Surrey Satellite Technology Ltd (SSTL) with the Disaster Monitoring constellation or the RapidEye constellation. Moreover, the solution envisaged needs to be highly responsive to the requests coming from the user community. The demonstrator of the DAFA study (Ocon et al. 2008) aims at addressing these issues with a multi agent architecture based on negotiation paradigm and deliberative agents. However, the main limitation of this approach is in the lack of scalability and flexibility.

The EO scenario presented above motivates the investigation of the multi agent paradigms in the context of distributed platforms, to model the coordination and control aspects of such missions and to use them as the skeleton of a ground-based mission planning system. Moreover

*c.iacopino@surrey.ac.uk.
†p.palmer@surrey.ac.uk.
‡nicola.policella@esa.int.
§alessandro.donati@esa.int.
¶a.brewer@sstl.co.uk.

extending these systems with natural-inspired techniques can result in high adaptability and scalability. Section 1 introduces the multi agent paradigm, a general framework for modelling distributed system, before focusing on the self-organizing properties which such systems can present using a natural-inspired paradigm based on the ant colonies. Section 2 describes the case study considered, the Disaster Monitoring Constellation operated by SSTL. Section 3 presents the system developed whereas Section 4 shows some preliminary results of such a system applied to the case study. Finally, Section 5 discusses the main benefits and drawbacks of our approach. This work is based on previous papers (Iacopino et al. 2012) where we introduced the planning problem of an EO constellation and we addressed a static and a dynamic single platform case study. This paper represents a step forward along the same roadmap, addressing now a multiple platform case study.

1 Technical Background

As explained in the introduction, the majority of the new studies on planning for multiple spacecraft has adopted the multi agent paradigm to model the coordination and control aspects of such missions. This paradigm offers higher level of responsiveness and adaptability than a monolithic architecture. Multi Agent Systems (MAS) is a relatively new field bringing together techniques and theories from multiple disciplines. When multiple agents coordinate together for a common purpose, there are a number of different mechanisms that can be used. These approaches are strictly connected with the capabilities of the agents, which range across the spectrum from reactive to deliberative architecture. In essence, we can talk about performing a task in a highly planned manner (deliberative), or relying instead on an instantaneous spontaneous manner (reactive).The reactive approach is highly suited with problems with uncertainty. It is the most suitable for describing natural complex systems with high number of entities interacting with complex dynamics.

Self-Organizing Systems

Discussions on reactive behaviours naturally lead on to the concept of self-organization. In a system consisting of a large number of entities, the result of combining simple behaviours at local level can end in a spontaneous complex behaviour at the system level able to achieve significant results. Moreover, the structures or patterns exhibited at system level can be achieved in a self-organizing manner, without a central or external authority. As presented, self-organization is a desirable characteristic which need to be imported in artificial systems that cope with high uncertainty and dynamic environments, such as space applications. The challenge in designing a self-organizing system is that there is no systematic way to formulate required micro-level behaviours given desired top-level macro behaviours. Researchers have been experimenting with several mechanisms leading to self-organizing phenomena (Serugendo, Gleizes, and Karageorgos 2006). The most promising is the *stigmergy* mechanism, an indirect communication mechanism used by a number of insect colonies; the most common paradigm is the ant colony pattern able to achieve complex system behaviours.

Ant Colony Paradigm

Deneubourg (Deneubourg et al. 1990) demonstrated how the Argentine ant was able to choose successfully the shortest between the two paths to a food source. From there, Dorigo already in the early ’90s (Dorigo, Maniezzo, and Colorni 1996) developed a heuristic inspired on such a model, called Ant Colony Optimization (ACO). Nowadays ACO is a family of stochastic techniques for solving combinatorial optimization problems reduced in finding good paths through graphs. The inspiring idea is that the ants looking for food deposit pheromones along the path. These pheromones influence the following ants to get the same path. However only the shortest path will end having the strongest pheromone distribution because is the one that requires the minimum travelling time. This is an example of self-organizing problem solving strategy. The best path is expected to emerge with the strongest pheromone distribution. The uniqueness of the ACO algorithms is their constructive nature, as opposed to local search; they generate solutions adding solution’s components iteratively until completion. Without going in details, ACO algorithms present a number of engineering benefits such as scalability, robustness and adaptability and have been successfully applied to a wide spectrum of theoretical and real problems: routing such as the travelling salesman problem (TSP), assignment, subset such as the Knapsack problem and scheduling, the closest to the mission planning problems (Merkle, Middendorf, and Schmeck 2000; Huang 2001; Gravel, Price, and Gagne 2002; Chen et al. 2010). Moreover this paradigm has been extended and used as the coordination infrastructure for multi agent systems applied to industrial applications, called synthetic ecosystems.

Synthetic ecosystems

Multi agent systems called synthetic ecosystems aim at providing practical engineering solutions of industrial strength, exploiting the underlying logic of self-organizing systems natural-inspired (Brueckner 2000). Brückner showed how to develop a manufacturing system based on a pheromone field similar to the one used in the ant colony pattern. He represented the system as a network where the industrial machines and workpieces are single agents which propagate their intentions, represented as pheromones, downstream the network while resource agents propagate load forecasts, represented as pheromones, upstream. Several other works (Hadeli et al. 2004; Valckenaers, Kollingbaum, and Van Brussel 2004; De Wolf and Holvoet 2007) showed self-organizing manufacturing system using artificial ants which navigate through a number of pheromone layers. A similar idea has been used in the on-board coordination system for cluster of satellites developed by Tripp and Palmer (Tripp and Palmer 2010) where stigmergy was able to reduce the computational and communication overhead and the task duplication.

The system presented in this paper is based on the ant colony pattern; Section 3 is going to explain how this pattern has been extended to coordinate a multiple platform scenario.

2 Case study

The scenario considered is the Disaster Monitor Constellation (DMC). This platform is the first Earth observation constellation of low cost small satellites; it provides daily images for a wide range of applications, commercial or of public interest including disaster monitoring. The DMC satellites are designed and built by a UK company, SSTL. The constellation is currently composed of 6 satellites, flying at about 700 km of altitude, (Beijing-1, NigeriaSat-1, UK-DMC-2, Deimos-1, Nigeriasat-NX, Nigeriasat-2) owned by different entities. DMC works within the International Charter "Space and Major Disasters" to provide free satellite imagery for humanitarian use, in the event of major international disasters. The national civil protection authorities of Algeria, China, Nigeria, Turkey and UK are direct authorised users of the Charter. The problem of imaging campaign planning & scheduling for this constellation rises because the number of requests and the typology of customers that such a platform has to satisfy is quite varied and exceeds the capabilities of the whole system. The challenge is in giving the ability to the constellation to respond in reasonable time to a number of users, making asynchronous requests. As the problem considered is a planning problem, the terms *plan* and *solution* are going to be used without distinctions.

The costumers request images of specific targets within certain time windows. Because of the limited memory on-board, time constraints between requests and limited number of downlink passes, it is required to determine a subset of such requests which satisfy all the constraints and maximize certain performance metrics. Given this context, the requirements for the mission planning system (MPS) go along three different dimensions:

- **Efficiency**, it needs to produce solutions that maximize the performance and minimize the duplications among the plans of each spacecraft.
- **Adaptability**, it needs to respond and adjust the solution when changes occur (new user requests, disaster management).
- **Scalability**, it needs to be scalable on the number of requests and spacecraft considered.

The challenge is to build a system that satisfies all these requirements at the same time. They are often in contrast; a system very adaptable is often not very efficient and vice versa. The requirement of coordination described in the introduction is a critical aspect for the system. It affects the scalability in terms of the spacecraft considered because it influences the number of possible images duplicated among their plans. These duplications have an impact on the efficiency of the entire system because they represent low resources' utilization. A possible duplication happens when two satellites can image the same target in a time frame considered too short for the customer of that target. Duplications therefore need to be defined by the operator that knows the revisit time requested by the customers. For targets at high priority, such as disaster management, we usually need many images in a very short time frame. In the case of a constellation with satellites lying on the same orbital plane, the ground tracks intersects only at the polar regions in the time-frame of a single orbit. However, considering a time frame of multiple days, several duplications can occurs because they dependent on the constellation revisit time that for a constellation such as DMC is daily. Moreover, they cannot be handled with a static planning system because they change at the same rate of the customer requests. Considering constellations, with heterogeneous satellite orbiting in different orbital plane the situation is even more complex and makes harder extending the planning horizon.

The following section introduces our system aiming at matching the requirements presented above.

3 Proposed approach

The MPS we aim to build is focused on the customers requests and, in the EO scenario considered, the level of uncertainty is quite low and the communication link is not a critical resource. The system is therefore foreseen to run centrally on the ground segment, abstracting from on-board processing and communication aspects among the satellites.

Differently from the standard operational workflow, our approach is going to run continuously offering an updated plan at any time. Traditionally, the plan is generated only for a specific uplink opportunity and when all the input are available. In our approach instead, the operator is called to evaluate a number of equivalent plans proposed by the system during a specific time frame. The system acts as an interface abstracting the decisional task of the operators from the problem variability. This setup offers a higher flexibility for the operators and a higher responsiveness to asynchronous events. However, these advantages need to match the resources available in terms of uplink opportunities and manpower dedicated to the plans revision.

The system we propose is inspired by self-organizing multi agent architectures, as defined in Section 1. Inside this field, we target the ant colony pattern able to achieve high-level of performance in optimization problems and high-level of scalability. To apply this paradigm we need to represent our planning problem as a graph-like environment, which ant-like agents can explore. Broadly speaking, we aim at implementing an MPS that behaves as an ant colony, continuously exploring and exploiting its environment, which represents the planning problem, and adapting to its changes.

In this section we first examine the problem representation, translating the planning problem in a graph-like environment. We then focus on the logic of the ant colony algorithm that allows a single spacecraft to optimize its plan and to adapt to the environment's changes.

Lastly, we present how this paradigm can be extended to offer a self-organizing coordination system for EO constellations.

Problem representation

Considering a single spacecraft, the problem domain can be modelled as a binary reusable resource, the spacecraft camera, strictly dependent on a depletable resource, the spacecraft on board storage memory. An image request is an activity that consumes memory while locking on the camera, we call these activities tasks. The ground station passes allow to download data, they can be modelled therefore as activity that produces memory. All these activities are on the camera timeline and are subjected to memory availability constraints and temporal constraints. The tasks are characterized by the memory needed and the *priority* which indicates the importance of the specific task; this last parameter is the results of a number of factors such as customer priority, weather forecast, rolling angle and so on. The ground station pass is indicated only with the memory that can be downloaded. That being defined, the problem can be represented as a knapsack problem with scheduling constraints. It can be formulated as:

$$\max \quad Q(\bar{X}) \tag{1}$$

subject to

$$\sum_{i=1}^{n} r_i x_i \leq a, \tag{2}$$

where $\bar{X}$ is a vector of $x_i \in \{0,1\}, \quad i = 1 \dots n$, x_i is an assignment variable that indicates if the request *i* has been performed. Equation (1) is a generic objective function that need to be maximised, taking in account the tasks selected and their relative priorities. This function represents the quality of a plan. Lastly, eq. (2) expresses the memory constrains.

The most natural way to translate a knapsack problem to a graph is to use a binary representation inspired by the assignment variables itself which characterize the knapsack problem (Kong and Tian 2005; Wei, Tuo, and Jing 2010; Fernandes, Ramos, and Rosa 2007). In this problem the variables are binary and the two possible states can be represented as distinct edges. In conclusion, the solution is just the path connecting these edges. From here on the terms *path* and *solution* are going to be used without distinctions. Figure 1 shows the binary representation of the problem where the squares represent the task and the triangle the ground station passes. This representation is also convenient for dynamic problems as it offers the possibility to express any events with minor changes in the graph reducing the overall impact. A pre-processing phase mission/problem specific is necessary for this translation. We are not interested in instrument operational details or in the spacecraft maneuvering model because we are considering agile satellites with negligible slewing and setup time.

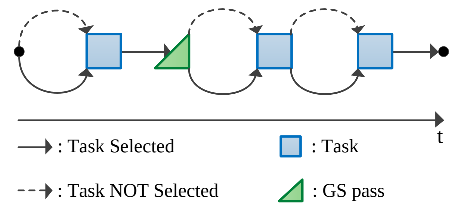

Figure 1: Problem representation as binary chain.

Ant colony

This subsection is focusing in the system's dynamics, characterized by distinct behaviours at local and global level. At local level, the ant-agents are inspired by the ACO paradigm, described in Section 1. Each ant, navigating the environment, the binary chain described above, is statistically driven by the pheromones found along each edge of this graph; the pheromones represent the recent history of the decisions of the previous ants. The objective function influences the quantity of pheromones, which the ant deposits on its path. Thanks to this coupling between ant decisions and objective function, the colony collectively optimizes the objective function. In our approach, the ants start their path always from the beginning of the planning windows and have to respect the time direction, which reflects the order of the tasks positioned in the ground track. The ants must separately check inconsistencies on the memory utilization. The ant workflow can be summarized as follows:

1: **Construction Phase**: the ant decides its path using a probabilistic rule, function of the pheromone trail;
2: **Objective Function Evaluation**: the path quality is determined using the objective function;
3: **Depositing Phase**: the ant deposits on its path an amount of pheromones, function of the path quality;

Considering the colony as a whole, the global behaviour of the system is different from the one of a specific ant. The key element of this algorithm is that the solution of the planning problem is not the solution found by one specific ant but the results of the interaction of all of them. After a certain number of ants have navigated and deposited, the pheromone trail reaches levels allowing almost the entire colony to repeat always the same path. This is regarded as a global solution. We are interested in developing a system that continuously adapts its current solution without knowledge on when a change occurs. To achieve this, every time a global solution is obtained, the colony is forced to leave that solution and to restart the exploration. The system execution therefore can be regarded as a repeating cycle of exploration, when the ants are pushed to find new solutions, and exploitation when the colony converges on one of them. The novelty of our approach is that this cycle is regulated by a controller parameter able to affect the stability of the long-term behaviours of the colony dynamics. Thanks to this parameter the colony is forced to converge and to leave again for exploration in a predefined time-frame. We have developed a theoretical model that can describe and foresee the long-term system's dynamics. This theoretical model gives us confidence in the system's reliability. A strong model is a priority for a system applied to critical scenarios such as mission planning. Further details on the theoretical model can be found in (Iacopino and Palmer 2012). The colony workflow can be summarized as follows:

```
PheromoneTrailInitialization();
ControllerParameterInitialization(), start exploration
phase;
for all ant do
    Construction/Depositing phase;
    if Colony converged then
        SaveGlobalSolution();
        ControllerParameterInitialization(), restart
        exploration phase;
    else
        UpdateControllerParameter();
    end if
end for
```

The *UpdateControllerParameter()* operation modifies the controller parameter using a function that progressively reduces the ants exploration while increases their exploitation. This type of algorithm is developed for dynamic problems exploiting the system's dynamics occurring in this type of graphs. An extended testing phase showing the benefits of this approach can be found in (Iacopino et al. 2013).

We envisage ant agents exploring continuously the environment, changing continuously the pheromone distribution. The ground segment is going to update at any time the environment with new information coming from the users, the spacecraft or from the real environment (weather forecast). At every ground station pass, the spacecraft can receive the current plan corresponding to the pheromone trail with the higher level of pheromones.

Coordination mechanism

The previous subsection was able to show the mechanisms behind the high-level of adaptability and efficiency of our system. However a further step is necessary to demonstrate its highly scalability. This subsection aims at drawing the general picture of our approach. The goal is to avoid duplications of the images acquired among the satellites and at the same time to optimize the performance of each spacecraft. Taking inspiration by the synthetic ecosystems seen in Section 1, each spacecraft is associated to an ant colony in charge of navigating a graph representing the planning problem of that spacecraft. These graphs are modelled as binary chains as explained above. The tasks shared among the satellites, representing possible duplications, are modelled as intersections among the satellites' binary chains. Figure 2 explains this representation.

To achieve coordination on the shared tasks, we exploit the pheromone fields generated by the ant colonies. We introduce a coupling similar to the one seen above between the ant decisional process and its deposit activity. In this case, we add a further link between the ants' deposit activity

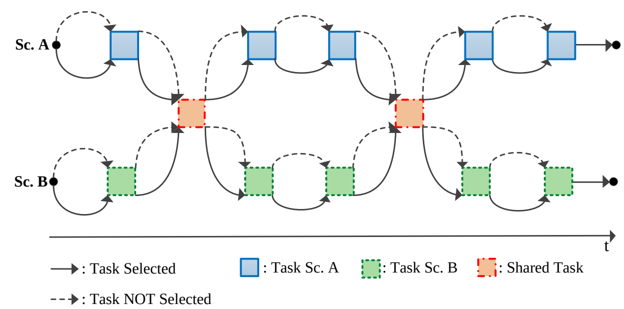

Figure 2: Problem representation in case of multiple spacecraft with shared tasks.

of one colony with the ants' decisional process of the colonies that share the tasks. Basically, when the ant of one spacecraft decides to perform a shared task, concurrently with the ants of the others spacecraft, it deposits pheromones on its path and also on the edges of the binary chains intersecting that task. Specifically, the ant will deposit only on the edges that inhibit the decision of choosing that task for the others colonies. This simple mechanism guarantees the coordination among the colonies, i.e., among the satellites, in a highly scalable manner. This approach does not have single point of failures; a common limitation of the hierarchical coordination systems. Our theoretical model has been extended to incorporate this mechanism and to confirm its validity. Further details on the analytical model are outside the scope of this paper.

The following section presents a qualitative analysis used to demonstrate empirically the validity of our approach.

4 Empirical evaluation

This section shows how the system operates with real-instance problems. We analyse two problems, aiming at demonstrating the adaptability and scalability features of the system. For sake of clarity, the first problem focuses only on the adaptability aspect while the second one only on the scalability feature. Rather than showing batch performances, we believe is more meaningful to give an intuitive understanding of what the system does during a single run.

Dynamic scenario

The scenario considered here is a dynamic problem for one spacecraft. This scenario is useful to understand how the system works for a single spacecraft and how it adapts to a changing problem. The setup of the experiment sees the system running for a long time frame. The time is measured in number of ants, because the graph is explored and modified only by one ant at a time. New events, i.e. changes in the problem are translated to changes in the environment, i.e. in the graph.

We consider two typologies of events, which define different type of changes:

- **Weather updates**, the weather information is a key factor

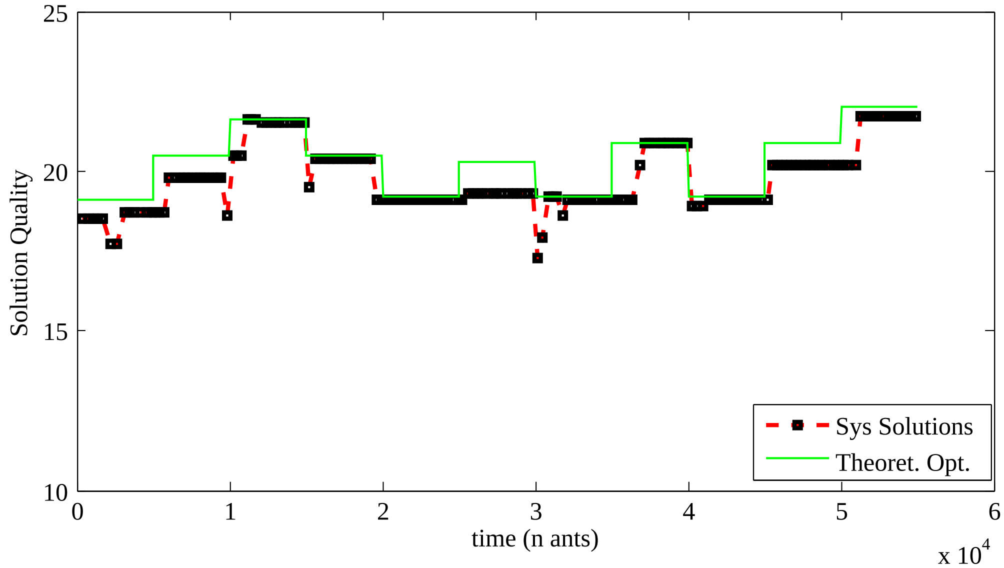

Figure 3: Quality evolution of the solutions calculated by the system.

for the image requests. Update weather information need to be taken in account to realize an efficient plan. On the graph, this information affects the tasks' *priority*, which translates in the amount of pheromone deposited on the relative path.

- **Disaster management**, new images at high priority can be requested at any time. In this case, the images are translated to new tasks, which need to be inserted in the graph. It is important to note that the impact on the graph structure is minimal.

The problem considered is formed by 20 possible tasks with 5 ground station passes. The chain's length depends on the number of requests. We can consider chains of any length. In this case, these numbers can represent a typical planning horizon of one day. Of note is the download capacity for each ground station pass is inferior to the whole memory onboard. Given a dynamic problem, we are interested in observing how the system responds to such changes. Moreover as explained in Section 3, independently from any change, the system continuously searches for new solution and updates the current plan. Figure 3 shows the result of one run where the system operates in a time frame of 40k ants. During this time the problem experiences 10 changes, reflected in variations in quality of its theoretical optimum, the green continuous line. The optimum is calculated off-line using a complete algorithm. The y-axis shows the quality value of the objective function as defined by eq.(1). The x-axis is the computational time measured in number of ants; the real time will depend on the computational capabilities of the ground segment. At this stage, the computational power required is negligible as the whole simulation takes few seconds in a desktop pc quad-core. Each point of the dotted line represents a solution calculated by the system. It is possible to note that in most of the cases the system finds the optimum or experiences fluctuations from it of about 10% of the solution's quality. A quantitative analysis on the adaptability properties of the algorithm proposed can be found in (Iacopino et al. 2013). Moreover, in this anaysis we show that the system offers always similar solutions in terms of decisions. It is important that the current solution deos not change completely at every event. This system's behaviour is given by the way how the exoloration of new solutions is performed. The exploration starts from the previous solutions and considers first its neighbours.

Constellation scenario

In this subsection we analyse a constellation scenario. The problem considered is static and is formed by 3 spacecraft, each with 20 possible tasks and about 5 ground station passes. Each spacecraft has 4 shared tasks with the others for a total of 6 possible duplications. In this paper, we are not interested in details regarding the spacecraft's orbits because we envisage a pre-processing phase where this information shall define the shared tasks. The experimental setting sees the comparison of the self-organizing system proposed by this paper and the same system without the self-organizing mechanism responsible of the spacecraft's coordination. Without this mechanism, the plan generated maximizes the local spacecraft plans without avoiding duplicated tasks.

The following charts are the results of one run where the system operates in a time frame of 20k ants on the problem defined above. Figure 4 shows the amount of duplicated tasks observed along the time. The x-axis represents a computational time measured in number of ants. Of note is that the self-organizing mechanism never shows any duplicated task. Moreover, the computational power required grows linearly with the number of spacecraft considered, confirming the scalability property of the self-organizing mechanism.

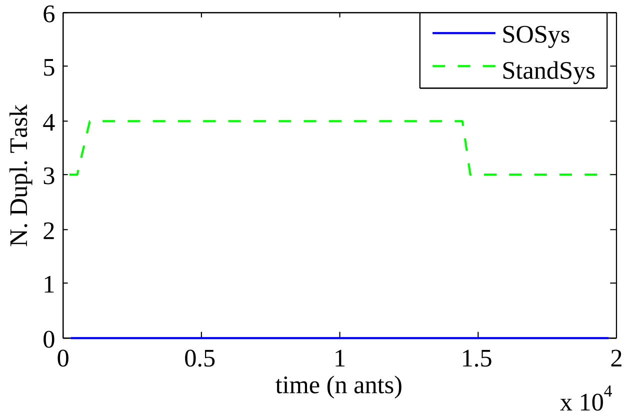

Figure 4: Evolution of the number of duplicated tasks with or without the self-organizing mechanism.

It is important to analyse the impact of duplicated tasks on the system's performance. Instead than analysing the solutions' evolution of each spacecraft, as seen in the previous subsection, we prefer to show metrics relevant to the entire constellation. We define therefore the following constellation plan's quality as:

$$Q_c(s) = \sum_{i=1}^{n} Q_i(\bar{X}_i) - D(\bar{X}_d) \quad (3)$$

where s is the system's solution for the constellation plan and n is the number of spacecraft considered. Equation (3) is formed by a first term that takes in account the quality of the plan of each spacecraft not caring of possible duplications with other spacecraft and by a second term that takes in account the duplicated tasks, indicated by the vector $\bar{X}_d$. Given this definition, Figure 5 show on the evolution of the quality of the solutions generated for the entire constellation. This figure, comparing the two systems defined above, highlights the benefit of the coordination mechanism over the standard system.

Analogous results can be observed analysing the system's efficiency expressed in terms of onboard storage memory utilization. In case of duplications a certain amount of memory becomes useless causing a decrease of efficiency. Figure 6 shows the evolution of the efficiency along the run.

The analysis presented here clearly shows the benefits of the system proposed in terms of efficiency, adaptability and scalability. However it does not offer quantitative measures and we are not comparing this method with other techniques. A complete analysis of the system's performance is therefore the next step or our research.

5 Conclusions

Distributed missions present new challenges for automated systems. They need to be highly responsive, adaptable

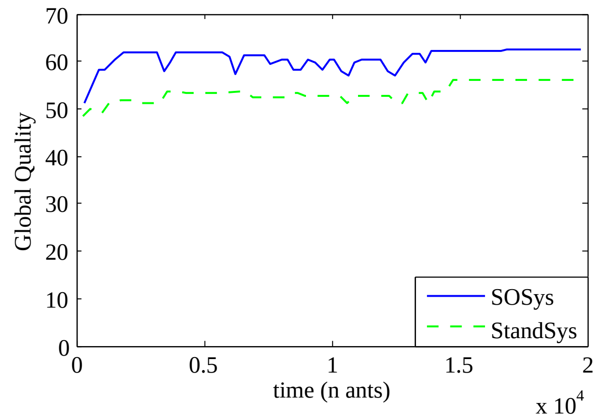

Figure 5: Quality's evolution of the constellation plan calculated by the system with or without the self-organizing mechanism.

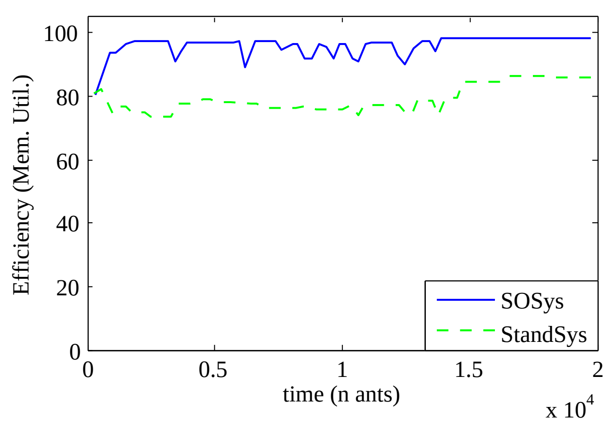

Figure 6: Efficiency's evolution of the constellation plan calculated by the system with or without the self-organizing mechanism.

to face dynamic environments and scalable in terms of number of spacecraft and image requests considered. Today a number of advanced technologies are available for meeting these requirements such as self-organizing multi agent architectures and natural-inspired collective algorithms. In this paper, we presented a system based on these technologies to face distributed missions' scenarios. In this paragraph, we want to summarize the main benefits and limitations of our approach. The main benefits are:

- **Efficiency**, the system developed exploits the optimization capabilities of the ant colony paradigm, a technique able to achieve high performance in a number of contexts, in particular in assignment and scheduling problems.
- **Adaptability**, self-organizing multi agent architectures are by definition more flexible and adaptable then

monolithic systems. The system developed, thanks to the ant colony paradigm, is highly adaptable due to the integration of the problem dynamics in the solution construction.

- **Scalability**, classic multi agent architectures suffer of scalability due to the strong responsibility schema. A self-organizing approach offers scalability at a reasonable price in terms of efficiency.

Despite these benefits, a number of limitations need to be taken in account:

- **Problem modelling**, the planning & scheduling problem need to be translated to a graph-like structure. This formalism cannot represent all the types of constraints but they can be incorporated in the agents' logic. A pre-processing phase mission/problem specific is necessary for this translation.
- **Black box**, all the *soft-computing* techniques such as neural networks and ant colony algorithms offer solutions without showing the relative reasoning chain. This is a critical issue from the human operators' prospective. However, the operator needs to become a supervisor and needs powerful tools for this task. The system presented in this paper looks in this direction giving the operator the possibility of choosing among a number of equivalent solutions.
- **Cost & Operational feasibility**, we are proposing a different operational workflow. As explained in Section 3 the system generates continuously plans of equivalent quality which the operator is called to evaluate. Such a system offers more flexibility but it requires a different manpower management.
- **Stochastic nature**, a further challenge is the mindset shift. The issue is to shift from a deterministic to a stochastic system. This is a mandatory step to build more complex and adaptable systems.

Acknowledgments

This work is co-funded by the Surrey Space Centre (SSC) of the University of Surrey, the Surrey Satellite Technology Ltd (SSTL) and the Operations Centre of the European Space Agency (ESA/ESOC).

References

Brueckner, S. 2000. *Return from the ant. synthetic ecosystems for manufacturing control*. Ph.D. Dissertation, Humboldt-UniversitŁt, Berlin.

Chen, W.; Zhang, J.; Chung, H.; Huang, R.; and Liu, O. 2010. Optimizing discounted cash flows in project scheduling - an ant colony optimization approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40(1):64–77.

Chien, S.; Cichy, B.; Davies, A.; Tran, D.; Rabideau, G.; Castano, R.; Sherwood, R.; Mandl, D.; Frye, S.; Shulman, S.; Jones, J.; and Grosvenor, S. 2005. An autonomous earth-observing sensorWeb. *IEEE Intelligent Systems* 20(3):16–24.

Chien, S. A.; Doubleday, J.; McLaren, D.; Tran, D.; Tanpipat, V.; Chitradon, R.; Boonya-aroonnet, S.; Thanapakpawin, P.; Khunboa, C.; Leelapatra, W.; Plermkamon, V.; Raghavendra, C.; and Mandl, D. J. 2011. Combining space-based and in-situ measurements to track flooding in thailand. In *IGARSS'11*, 3935–3938.

De Florio, S. 2006. Performances optimization of remote sensing satellite constellations: a heuristic method. In *International Workshop on Planning and Scheduling for Space*.

De Wolf, T., and Holvoet, T. 2007. Designing Self-Organising emergent systems based on information flows and feedback-loops. In *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on*, 295–298.

Deneubourg, J. L.; Aron, S.; Goss, S.; and Pasteels, J. M. 1990. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior* 3(2):159–168.

Dorigo, M.; Maniezzo, V.; and Colorni, A. 1996. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26(1):29–41.

Fernandes, C.; Ramos, V.; and Rosa, A. C. 2007. Stigmergic optimization in dynamic binary landscapes. In *Proceedings of the 2007 ACM symposium on Applied computing*, 747–748.

Grasset-Bourdel, R.; Verfaillie, G.; and Flipo, A. 2011. Building a really executable plan for a constellation of agile earth observation satellites. In *IWPSS - International Workshop on Planning & Scheduling for Space*.

Gravel, M.; Price, W. L.; and Gagne, C. 2002. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research* 143(1):218–229.

Hadeli, K.; Valckenaers, P.; Zamfirescu, C.; Brussel, H. V.; Germain, B. S.; Hoelvoet, T.; and Steegmans, E. 2004. Self-organising in multi-agent coordination and control using stigmergy. *Engineering Self-Organising Systems* 105–123.

Huang, S. J. 2001. Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches. *Energy Conversion, IEEE Transactions on* 16(3):296–301.

Iacopino, C., and Palmer, P. 2012. The dynamics of ant colony optimization algorithms applied to binary chains. *Swarm Intelligence* 6(4):343–377.

Iacopino, C.; Palmer, P.; Policella, N.; Donati, A.; and Brewer, A. 2012. Highly responsive MPS for dynamic EO scenarios. In *The 12th International Conference on Space Operations, SpaceOps 2012*.

Iacopino, C.; Palmer, P.; Brewer, A.; Policella, N.; and Donati, A. 2013. A novel aco algorithm for dynamic binary chains based on changes in the system's stability. In *2013 IEEE Swarm Intelligence Symposium (SIS-13)*. Piscataway, NJ: IEEE Press. TBD.

Kong, M., and Tian, P. 2005. A binary ant colony optimization for the unconstrained function optimization problem. *Computational Intelligence and Security* 682–687.

Lenzen, C.; Worle, M.; Mrowka, F.; and Geyer, M. 2011. Automated scheduling for TerraSAR-X/TanDEM-X. In *IWPSS - International Workshop on Planning & Scheduling for Space*.

Mandl, D.; Sohlberg, R.; Justice, C.; Ungar, S.; Ames, T.; Frye, S.; Chien, S.; Tran, D.; Cappelaere, P.; Sullivan, D.; and Ambrosia, V. 2008. A space-based sensor web for disaster management. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008.*, volume 5, V – 294–V – 297. IEEE International.

Merkle, D.; Middendorf, M.; and Schmeck, H. 2000. Ant colony optimization for Resource-Constrained project scheduling. *IEEE Transactions on Evolutionary Computation* 6:893–900.

Ocon, J.; Rivero, E.; Sanchez Montero, A.; Cesta, A.; and Rasconi, R. 2008. Multi-agent frameworks for space applications. In *SpaceOps 2010*.

Pralet, C.; Verfaillie, G.; and Olive, X. 2011. Planning for an ocean global surveillance mission. In *IWPSS - International Workshop on Planning and Scheduling for Space*.

Raghava Murthy, D. A.; Kesava Raju, V.; Srikanth, M.; and Ramanujappa, T. 2010. Small satellite constellation planning for disaster management. In *International Astronautical Federation*.

Serugendo, G. D.; Gleizes, M. P.; and Karageorgos, A. 2006. Self-organisation and emergence in MAS: an overview. *Informatica* 30(1):45–54.

Tripp, H., and Palmer, P. 2010. Stigmergy based behavioural coordination for satellite clusters. *Acta Astronautica* 66(7-8):1052–1071.

Valckenaers, P.; Kollingbaum, M.; and Van Brussel, H. 2004. Multi-agent coordination and control using stigmergy. *Computers in Industry* 53(1):75–96.

Wang, P., and Tan, Y. 2008. Joint scheduling of heterogeneous earth observing satellites for different stakeholders. In *SpaceOps 2008*.

Wei, K.; Tuo, H.; and Jing, Z. 2010. Improving binary ant colony optimization by adaptive pheromone and commutative solution update. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 565–569. IEEE.