JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 46, No. 3, May–June 2009

Science Operations Planning Optimization for Spacecraft Formation Flying Maneuvers

R. Burgon* and P. C. E. Roberts†
Cranfield University, Cranfield, Bedfordshire MK43 0AL, United Kingdom
J. A. Roberts‡
Aon Space, London EC2M 4PL, United Kingdom
and
F. Ankersen§
ESA, NL-2200 Noordwijk, The Netherlands

DOI: 10.2514/1.40490

The use of formation flying of distributed space systems, especially for space-based interferometry, is receiving much attention by mission designers. Optimal maneuver planning for these missions is critical to ensure the safe operation of the spacecraft and to maximize the mission science returns. One such mission, Darwin, requires complex observation scheduling complicated by a number of interconnected temporal constraints placed on the mission. Following a defined maneuver planning architecture this paper introduces a science operations planner that helps maximize observation time through science operations schedule optimization. Comparison of this method with a simple benchmark planner is given and shows that schedule performance increases of up to 12% can be achieved. Though these increases can only be achieved using significantly more computational resources than the benchmark planner, they are found using constraints that would allow the planner to be able to operate autonomously onboard one of the formation spacecraft.

Nomenclature

a = vertex index number $\mathbb{N}$: $\{1, \ldots, k\}$
c_j = task completion duration for task j, days
D = mean task duration of a tour, days
E = set with infinite number of elements and range $\mathbb{R}$: $\{0, \ldots, 3\}$ representing a number line
f = task prioritization flag $\mathbb{N}$: $\{1, \ldots, \infty\}$
I = number of iterations of the science operations planning algorithm
i, j = index numbers for sets $\mathbb{N}$: $\{1, \ldots, \infty\}$
k = number of vertices in U_i
L = tour duration, days
$l_{i,j}$ = duration of the vertex $\mathbf{v}_{\mathbf{i},\mathbf{j}}$, days
M_i = set of nodes $\subset N$ representing all achievable tasks starting from n_i
$m_{i,j}$ = maneuver duration from task i to task j, days
N = set of nodes representing all science tasks
n_i = node $i \in N$
$\dot{O}$ = mean rate of extra observation time gained using the science operations module instead of the benchmark planner to plan tours, h/day
O_i = set of nodes $\subset N$ representing all tasks within field of view defined by $\beta(t_i)$
p = number of vertices in T
p_{waits} = number of "wait" vertices in T
R = task/time ratio of the tour, tasks/day
r = random number generated from a normal distribution with zero mean and variable σ
R_{BP} = task/time ratio of a tour generated by the benchmark planner, tasks/day
R_{SOM} = task/time ratio of a tour generated by the science operations module, tasks/day
S = set of star positions represented in ecliptic longitude
$s(n_i)$ = ecliptic longitude of star associated with node n_i, °
T = set of vertices in tour
t_i = time of task completion for node n_i
T_{target} = user-defined minimum tour duration, days
U_i = set of all vertices from n_i to all nodes in M_i
V_i = set of all vertices from n_i to all other nodes
$\mathbf{v}_{\mathbf{i},\mathbf{j}}$ = vertex from n_i to n_j
$V_{i,\text{sort}}$ = set of vertices U_i sorted by increasing vertex duration $l_{i,j}$
V^* = set of vertices $V_{i,\text{sort}}$ arranged with prioritized vertices at the beginning
$\mathbf{v}_a^{*k}$ = vertex representing the ath element of V^*
W_i = set of vertices connecting n_i to $n_j \in O_i$
$\dot{\alpha}$ = angular rate of the field of view, deg/day
$\beta(t_i)$ = ecliptic longitude of antisun vector at time t_i, °
σ = standard deviation of the random number generator

I. Introduction

AUTONOMOUS formation flying of spacecraft to perform separated spacecraft interferometry is a theme for many space missions [1–4]. ESA's Darwin mission proposal uses four spacecraft flying in formation to perform interferometry on distant stars to identify the existence of extrasolar planets and analyze their atmospheres [5]. To ensure the accuracy and stability of the observations the spacecraft and formation guidance and control systems must be able to perform a number of different maneuvers including stationkeeping, formation-keeping, and reconfiguration maneuvers to a high accuracy. The planning of these maneuvers is crucial to ensure spacecraft safety and ensure the mission maximizes the science returns while minimizing fuel cost. This paper is concerned with the science operations scheduling side of maneuver planning and describes an approach to scheduling that can operate autonomously

Presented at the AAS/AIAA Astrodynamics Specialist Conference, Mackinac Island, MI, 19–23 August 2007; received 19 August 2008; revision received 3 March 2009; accepted for publication 6 March 2009. Copyright © 2009 by Cranfield Univ. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0022-4650/09 $10.00 in correspondence with the CCC.

*Ph.D. Candidate, Space Research Centre. AIAA Student Member.

†Course Director/Lecturer, Space Research Centre.

‡Aon Limited, 8 Devonshire Square.

§Control Systems Analyst, Guidance, Navigation, and Control Systems, PO Box 299.

onboard a spacecraft in real-time within a unified maneuver planning architecture.

The necessity of science operations planning can be emphasized by considering the Darwin mission. The goal of the mission is to identify extrasolar planets and analyze their atmospheres using interferometry from the vicinity of the sun/Earth L2 point. The target star catalogue contains approximately 450 stars.[¶] Each star requires at least three separate observations for planet detection and up to six additional analysis observations if a planet is detected. This potentially leads to over 4000 individual observations ranging in duration from $\frac{1}{3}$ day to approximately 85 days [6]. With a 5-year mission profile and to maximize the number of observations that can be completed within this time it is important to carefully plan the science operations schedule. A priori planning of the observation schedule for the entire mission cannot be accomplished, however, because there is no information regarding which stars within the mission star catalogue have planets. Therefore real-time scheduling is required as this allows for updated information to be included in the planning algorithm. There are a number of observation planning issues that must be addressed by this real-time algorithm that are specific to Darwin:

1) The formation must remain pointing within 45 deg of the antisun vector [6]. The field of view for the mission therefore is constantly rotating (at a rate of approximately 0.986 deg/day) and every star in that field of view remains there for approximately 90 days/year. The scheduling of observations on stars that leave the field of view before the observation is complete must be avoided.

2) Additional scheduling time constraints are also posed by the maneuver time required by the formation to slew between target stars and the calibration time required to correctly image the star.

3) Planet detection observations must be scheduled before atmosphere analysis observations.

4) Multiple detection observations are required to confirm planet detection [6]. These multiple observations must be separated by enough time so that any potential planet can be observed at different points around its orbit. For a sun/Earth-like system this could be as much as 90 days (quarter of an orbit).

5) Once the detection observations have been completed there will be analysis time required to confirm detection and determine the planet's orbital characteristics. This will delay the scheduling of the atmosphere analysis observations.

6) Depending on the orbital characteristics of the detected planet occultation by (or transit of) the parent star may occur. Atmospheric analysis of the planet during these times needs to be avoided further constraining the scheduling of observations.

7) For the number of stars in the catalogue and the number of separate observations that might be performed on them there is a high likelihood that completing all the observations will take much longer than the planned 5-year mission duration.

All these planning issues make science operations planning a complex problem, hereafter called the science task assignment (STA) problem (where a "task" is defined by a combination of the target star and the type of observation performed).

In this paper a science operations planning algorithm, the science operations module (SOM) is introduced and defined within a larger maneuver planning architecture. The SOM is compared with a simple benchmark scheduling algorithm and shown to be superior in planning observation schedules for a Darwin-like interferometry mission.

II. Previous Contributions

Planning and scheduling for autonomous spacecraft is a relatively new concept that has been put into practice in various forms on a number of space missions. These include Cassini [7], Rosetta [8], and Deep Space One [9]. A useful introduction to the planning systems used at NASA and specifically for Deep Space One and Earth Orbiter One from the New Millennium Program is found in [10]. It describes the benefits of ground-based and onboard operations planners as "reduced costs, increased responsiveness, increased interactivity, increased productivity, and simplified self-monitoring." The planners use "symbolic AI" (artificial intelligence) routines to review any given plan, identify the flaws, and iteratively remove the flaws to create a flawless plan. These papers, however, describe operations planners that take into account all of the subsystems and payloads on a spacecraft and find a schedule that allows the most efficient use of spacecraft resources. More recently an autonomous mission planning system was introduced called the Multi-Agent Planning System (MAPS) [11,12]. It is designed for a fully autonomous deep-space mission and combines the results from a number of planning agents (representing the spacecraft subsystems) with a planning manager agent to create mission operations schedules for the complete spacecraft. This is implemented in C + + and Java, respectively, but the authors only provide the architecture for the model, not the details of how the planner actually achieves its goals. Addressed in [11,12] is the problem of automating spacecraft operations; however, they do not address optimization within the planning environment. The tasks are scheduled within the time and resource allocation constraints defined, but no attempt is made to improve a plan once an achievable schedule has been found. This is different than the situation to be solved by the SOM. Because there will always be many more tasks available than can be completed in any defined time period, the SOM must optimize the schedule to increase the number of observations completed.

NASA's Starlight dual-spacecraft interferometry mission [1] is used as a baseline for examining interferometry mission schedules that reduce fuel expenditure and encourage fuel balancing between the spacecraft in a similar mission planning problem to that of the STA [13]. Fuel consumption dynamics for each maneuver are based on a free-space model described in [14]. In [13] maneuvers are separated into three types: retarget, resize, and reorientation. Each star, separated by retarget maneuvers, is imaged using a number of resize and reorientation maneuvers. The optimization of the maneuver schedule is likened to a travelling salesman problem (TSP). A "chained local optimization" (CLO) algorithm combining simulated annealing and local search optimization techniques is developed to solve the TSP. The authors used a benchmark tour as a comparison that involves completing all the resize and reorientation maneuvers for each star before retargeting to the next star. The results show that, for a number of different scenarios, the optimized tour is capable of reducing the fuel consumption of the spacecraft performing all maneuvers by a significant amount compared with the benchmark tour. This is achieved by combining resize, reorientation, and retarget maneuvers on multiple targets instead of performing all the resize and reorientation maneuvers on one target before moving to the next. These results are noteworthy in understanding the necessity for the SOM as an analogy can be made. In the paper, each star has a number of resize and reorientation maneuvers, each with their associated fuel costs. Likewise, the stars in the Darwin catalogue each have a number of observation tasks with their associated time costs. By extension it may be inferred that combining the science tasks for Darwin in an appropriate schedule is more time optimal than performing the observations in a more intuitive systematic fashion. There are a number of factors, however, that [13] does not address. The problem is time invariant and there are no pointing constraints. This means that a star's availability for observation is not included and the duration of each maneuver is omitted (because fuel consumption is the only factor being investigated). Furthermore, a fixed number of maneuvers are investigated, and each solution finds a tour that encompasses them all. As previously discussed, the SOM will have to find a tour with temporal constraints from a large number of possible tasks that will not all be able to be completed within the given time. Therefore, although the paper reveals some interesting trends with respect to interferometry mission scheduling, its relative simplicity prevents its methods from being used to emulate the SOM.

The mission scheduling problem can be likened to a TSP. In the basic TSP the salesman has to visit a number of cities, and the tour is optimal if the shortest route is chosen. The TSP and a variant, the vehicle routing problem (VRP), are cornerstone challenges within the field of operations research. Similar problems can also be found

[¶] den Hartog, R., private communication, May 2006.

in computer network routing literature. There are many different kinds of constraints that can be added to the TSP, including time windows (visiting a point within a specified time period) [15], path constraints (minimizing the number of times a separate point is crossed, e.g., a river) [16], or sequential constraints (visiting a point before another) [17]. These constraints can all be made applicable to the STA problem. There are also a vast number of different methods that can be employed to solve these problems. Exact algorithms like branch and bound, cutting plane, and linear programming are good for finding solutions with a relatively small number of points to visit. For larger problems, heuristics have been developed that quickly find good solutions (greater than 97% optimal). These include nearest neighbor [18], simulated annealing [16], genetic algorithms [19], and ant algorithms [20].

Another analogy is the single machine scheduling problem (SMSP) found in manufacturing. In this problem there is a list of n jobs (science tasks). Each job has a release time (the time the task enters the field of view), a duration (the observation time of the task), a deadline (the time the star leaves the field of view), and a weight (the importance of completing the task). The goal of any algorithm to solve this problem is to minimize the total number of late jobs (i.e., those completed after their deadline). Again, a number of conventional and innovative methods have been employed to solve this problem including branch and bound [21], local search [22], and memetic algorithms [23]. Both the TSP and SMSP represent problems in combinatorial optimization and are close analogies to the STA problem. There is one fundamental difference, however. The TSP and SMSP aim to find a schedule for a fixed number of cities/jobs. The STA problem, on the other a hand, involves a larger number of tasks than can be completed in the available time, and no task can be selected if it overruns its "deadline." This means that conventional combinatorial optimization methods like branch and bound, linear programming, and genetic algorithms cannot be used for solving the STA problem. Even if the optimization occurs over a fixed number of tasks (rather than over time), the pool of tasks that can be selected is far larger than the size of the set to be optimized. This also renders conventional combinatorial optimization techniques incompatible with the STA problem.

III. Maneuver Planning Architecture

The science operations planning algorithm presented in this paper is part of a larger maneuver planning architecture (MPA) [24] as shown in Fig. 1. The goal of the MPA is to optimally plan formation maneuvers for the mission including the selection of the target and observation, the requirement for station keeping (if any), the selection of the relative spacecraft positions that satisfy the observation goal, and the design of the trajectories to perform the maneuver. The MPA is designed to run during each observation phase of the mission to plan for the next maneuver phase (when the formation reconfigures to perform a different observation). Each stage of the MPA uses optimization to accomplish various goals including maximizing observation time, minimizing maneuver time, minimizing fuel consumption, fuel balancing throughout the formation, collision avoidance, and thruster plume impingement avoidance. The stage descriptions are as follows:

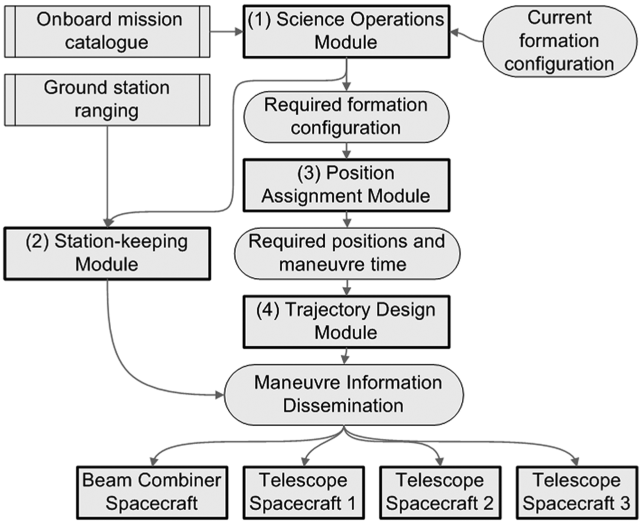
Onboard mission catalogue
(1) Science Operations Module
Current formation configuration
Ground station ranging
Required formation configuration
(3) Position Assignment Module
(2) Station-keeping Module
Required positions and maneuvre time
(4) Trajectory Design Module
Maneuvre Information Dissemination
Beam Combiner Spacecraft
Telescope Spacecraft 1
Telescope Spacecraft 2
Telescope Spacecraft 3

Fig. 1 Maneuver planning architecture.

1) The SOM, which contains the science operations planning algorithm, accesses the mission catalogue and current known formation configuration, and optimizes an observation schedule that is used to define the next science operation to perform.

2) The station-keeping module (SKM) uses the optimized schedule to calculate whether to perform a station-keeping maneuver during the next maneuver phase. If the next maneuver phase is selected for station keeping then the SKM outputs the station-keeping maneuver details. Otherwise there is no output from this module.

3) The position assignment module (PAM) [25,26] selects the first observation in the SOM's optimized schedule and cross-references it with the mission catalogue to obtain the formation configuration details that satisfy the observation requirements (e.g., pointing direction, baseline size, formation configuration, etc.). Then, using the current formation configuration details, it optimizes the final relative positions of the spacecraft to satisfy the desired configuration details. If a station-keeping maneuver is required then it is absorbed into the calculation. The outputs of this module are the spacecraft positions required to achieve the final formation configuration.

4) The trajectory design module (TDM) [26] uses the current formation configuration and the final formation configuration to optimize a collision and plume avoiding trajectory for each spacecraft to follow.

5) The final stage is the control stage and is not strictly part of the maneuver planning process. In the control stage each spacecraft follows their planned trajectories whereas a real-time collision monitoring algorithm is used to assess the accuracy of the trajectory following and intervene if deviation occurs.

Because the MPA is designed to run during every observation phase, the onboard catalogue can be updated from the ground as new information is made available. This means the SOM is likely to produce a different optimized schedule every time it is run, therefore keeping the target selection as optimal as possible.

IV. Problem Formulation

This section highlights the STA problem definition and introduces the SOM and the benchmark planner (BP) as possible solutions.

A. Problem Definition

Consider a set N of nodes, representing science tasks, and a set S of star positions, representing the ecliptic longitude of each star in the catalogue. Each pair of nodes $n_i \in N$ and $n_j \in N$ is connected by a vertex $\mathbf{v_{i,j}} \in V_i$, where the set V_i represents all possible vertices $\mathbf{v_{i,j}}$ connecting n_i to all other nodes. Relative weighting of the nodes with respect to the other nodes is given by the flag f ($f \in \mathbb{N}$: $\{1, \ldots, \infty\}$). If the node contains a flag (i.e., n_i^f) then the node (task) can gain priority scheduling with $f = 1$ the highest priority, $f = 2$ the next highest, and so on. If the node does not contain a flag (i.e., n_i) then there is no weighting for that task. Flags and their priorities can be added to any task throughout the mission by the science team to prioritize tasks within the schedule. M_i is a set of nodes ($\subset N$) representing all achievable tasks from node n_i connected by vertices in the set U_i ($\subset V_i$). A diagram of the nodes and sets defined thus far is given in Fig. 2, and the construction of M_i is detailed in the following.

Let $l_{i,j}$ be the duration (in days) of the vertex $\mathbf{v_{i,j}} \in V_i$ connecting n_i to $n_j \in M_i$

$$l_{i,j} = m_{i,j} + c_j \tag{1}$$

where $m_{i,j}$ is the maneuver duration and c_j the task completion duration. For this implementation of the SOM, $m_{i,j}$ is calculated from

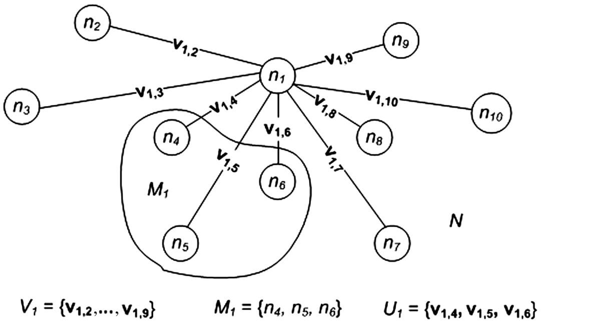

Fig. 2 Diagram of the node and vertex sets used in the SOM.

the angular separation of the initial star $s(n_i) \in S$ and the target star $s(n_j) \in S$ and a fixed formation angular rate $\dot{\alpha} = 0.02$ deg s^{-1} (this rate matches the average maneuver angular rate observed when maneuver planning using the PAM [25])

$$m_{i,j} = \dot{\alpha}|s(n_i) - s(n_j)| \tag{2}$$

If $s(n_i) = s(n_j)$ then $m_{i,j} = 0$.

The tour is a set T of p vertices $\mathbf{v_{i,j}} \in T$ of duration L where

$$L = \sum_{k=1}^{p} l_{i,j}^{k} \tag{3}$$

and

$$L \geq T_{\text{target}} \tag{4}$$

with T_{target} a user-defined minimum tour duration. The goal of the SOM is to find a tour that maximizes the number of vertices p within the minimum tour duration T_{target}. T_{target} can have a range from 0.116 days (the shortest task duration) to the entire mission lifetime of 5 years, however, the size of T_{target} will directly affect the length of time it takes to calculate a tour and the relative performance of the tour. For example, a combination of five $T_{\text{target}} = 4$ days tours will be quicker to calculate than one $T_{\text{target}} = 20$ days tour, but the latter will likely be a more optimal tour than the former. T_{target} is a minimum tour duration (as opposed to a maximum tour duration) so that tours can be generated that have a duration of at least T_{target}. If T_{target} was a maximum tour duration, the user would have less control over the length of the tour desired.

B. Science Operations Module Core Algorithm

At each node n_i, the set of nodes M_i is generated from which the set of vertices U_i can be calculated. This allows for the scheduling constraints detailed in Sec. I to be applied to V_i. M_i is found as follows:

1) Generate set $O_i \subset N$, representing all possible nodes with star positions $s(n_i) \in S$ such that

$$\beta(t_i) - \frac{\pi}{4} \leq s(n_i) \leq \beta(t_i) + \frac{\pi}{4} \tag{5}$$

where $\beta(t_i)$ is the ecliptic longitude of the antisun vector and t_i is the time when the node (task) n_i completes. O_i contains all the tasks for stars within the field of view bounded by ±45 deg from the antisun vector.

2) Calculate set W_i, representing all the vertices connecting n_i to $n_j \in O_i$.

3) To obtain the set of vertices U_i (and hence M_i) remove from the set of vertices W_i all the vertices where

$$n_i = n_j \tag{6}$$

and

$$s(n_j) \leq \beta(t_j) - \frac{\pi}{4} \tag{7}$$

where

$$\beta(t_j) = \beta(t_i) + l_{i,j}\dot{\alpha} \tag{8}$$

$s(n_j)$ is the star position for node n_j, $\beta(t_j)$ is the ecliptic longitude of the antisun vector, and t_i is the time when the node (task) n_j completes. The first condition, shown in Eq. (6), ensures that the task the formation is originally assigned to n_i is not included in the set M_i, thus preventing task duplication. The second condition, shown in Eq. (7), eliminates any tasks that cannot be scheduled as their star would leave the field of view before the task could be completed. After applying the constraints, U_i contains all the achievable vertices from n_i. The set of nodes M_i can be obtained by examining U_i and extracting all the nodes n_j.

4) If during step 3 $U_i \Rightarrow \emptyset$ then no suitable vertices exist and a wait vertex of 24 h in duration is imposed on the tour to allow $\beta(t_i)$ to move across the sky. The 24 h wait duration is chosen as it provides an approximately 72% chance that a new star will enter the field of view during the wait. If no new star enters the field of view during the wait period then another wait period is imposed on the tour. If two waits are required then the chance that a new star will enter the field of view during this second wait period rises to approximately 92%. The 24 h wait, therefore, provides a balance between a long enough duration to increase the chance that a new star will enter the field of view but not too long as to waste time. The preceding percentages are obtained by calculating the differences in ecliptic longitude of adjacent stars in the catalogue as a function of $\dot{\alpha}$ to find the field of view slew times between adjacent stars.

A diagram showing the set definitions for M_i and U_i is given in Fig. 3. In this example the set $N = \{n_1, \ldots, n_{10}\}$ is reduced to the set $O_1 = \{n_2, n_4, n_5, n_6\}$ through application of the field of view defined by $\beta(t_1)$, shown in Eq. (5). From O_1 the set $W_1 = \{\mathbf{v_{1,2}}, \mathbf{v_{1,4}}, \mathbf{v_{1,5}}, \mathbf{v_{1,6}}\}$ of vertices is calculated. Applying the conditions in Eq. (6) and (7) for node n_2 the vertex $\mathbf{v_{1,2}}$ is removed because n_2 is outside the $\beta(t_2)$ field of view. Assuming no violations of Eqs. (6) and (7) occur for nodes n_4, n_5, and n_6 the final vertex set is $U_1 = \{\mathbf{v_{1,4}}, \mathbf{v_{1,5}}, \mathbf{v_{1,6}}\}$, and, hence, $M_1 = \{n_4, n_5, n_6\}$ is obtained. M_1 contains all the nodes (i.e., science tasks) that can be completed from node n_1.

After U_i is calculated a decision process (described in the following sections) selects $\mathbf{v_{i,j}} \in U_i$, and the node is added to the tour. A file recording the status of each task (the taskflag file) is then updated, thus ensuring an up-to-date reference for the next iteration of the decision process. This process is repeated for n_j, etc., until $L \geq T_{\text{target}}$, at which point the tour is terminated. A flow diagram showing this shell algorithm can be found in Fig. 4. Vertex selection is the process that differs between the SOM and benchmark approaches to the STA problem.

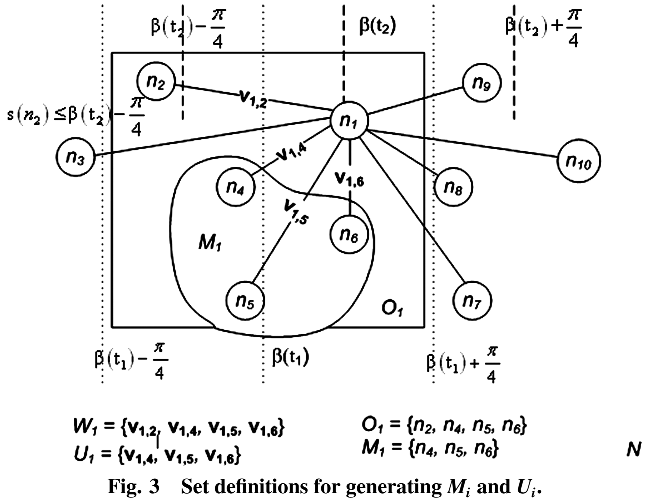
$\beta(t_2) - \frac{\pi}{4}$
$\beta(t_2)$
$\beta(t_2) + \frac{\pi}{4}$
n_2
n_9
$v_{1,2}$
$s(n_2) \leq \beta(t_2) - \frac{\pi}{4}$
n_1
n_3
$v_{1,4}$
n_{10}
n_4
$v_{1,6}$
n_8
$v_{1,5}$
n_6
M_1
n_5
O_1
n_7
$\beta(t_1) - \frac{\pi}{4}$
$\beta(t_1)$
$\beta(t_1) + \frac{\pi}{4}$
$W_1 = \{v_{1,2}, v_{1,4}, v_{1,5}, v_{1,6}\}$
$U_1 = \{v_{1,4}, v_{1,5}, v_{1,6}\}$
$O_1 = \{n_2, n_4, n_5, n_6\}$
$M_1 = \{n_4, n_5, n_6\}$
N

Fig. 3 Set definitions for generating M_i and U_i.

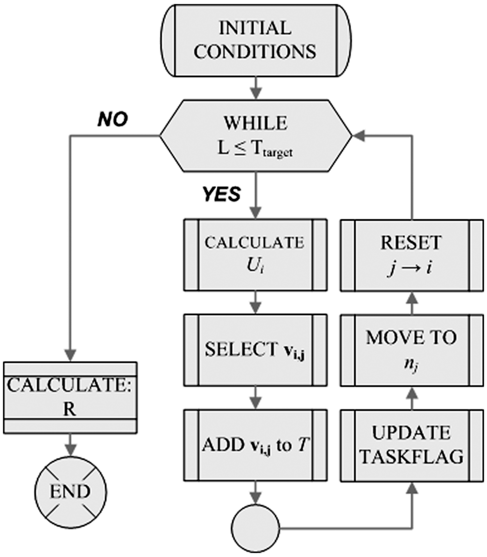

Fig. 4 Flow diagram for the core SOM algorithm.

The goal of the algorithm is to maximize the number of vertices (and hence the number of completed tasks) in the tour T. Because there is no fixed tour time (T_{target} only represents a minimum tour time) the metric chosen to rate the performance of a tour is the task/time ratio R:

$$R = \frac{(p - p_{\text{waits}})}{L} \tag{9}$$

where p_{waits} represents the total number of wait vertices imposed on the tour (see preceding step 4). In the tour performance calculation p_{waits} is removed from the total number of vertices p because the wait vertices account for scheduled time not allocated to science observations. The time signature of the wait vertices (24 h per wait) remains in the total tour time L because they negatively affect the performance of the tour. In evaluating the performance of a tour a higher task/time ratio indicates a better use of time within the tour.

C. Benchmark Vertex Selection Process

The benchmark vertex selection algorithm uses the principle of nearest neighbor selection [18]. The nearest neighbor here refers to nearest neighboring tasks as opposed to nearest neighboring stars. Here the nearest neighboring task is the one that takes the shortest time to complete. At each node one vertex needs to be selected from a group of all available vertices for that node (i.e., the set U_i). For the benchmark, the selected vertex is simply the one with the smallest completion time associated with it:

$$\mathbf{v_{i,j}} = \min_{l_{i,j}} U_i \tag{10}$$

The benchmark algorithm is "greedy" in nature in that it always selects the shortest vertex regardless of how it affects the tour later on. The vertex selection is locally optimal over one node but a tour constructed using the benchmark is unlikely to be globally optimal over T_{target} because the algorithm only examines a small section of the complete solution space. This method, however, is useful as a benchmark as it is easy to implement and provides solutions quickly.

D. Science Operations Module Vertex Selection Process

The SOM version of the vertex selection algorithm is a simplified version of the basic ant colony optimization metaheuristic [19]. As before, at each node one vertex needs to be selected from a group of all available vertices for that node (U_i). This vertex is chosen using a weighted stochastic process. First U_i is sorted in ascending order of $l_{i,j}$ such that

$$V_{i,\text{sort}} \in \{\min_{l_{i,j}} U_{i1}^k, \ldots, \max_{l_{i,j}} U_{ik}^k\} \tag{11}$$

Some of the nodes that the vertices in $V_{i,\text{sort}}$ point to may have prioritization flags (i.e., n_i^f). These vertices are moved to the beginning of $V_{i,\text{sort}}$ in order of prioritization to create the set V^*. Therefore V^* is the set U_i sorted into ascending order and rearranged to ensure prioritized vertices are at the beginning. $\mathbf{v}_a^{*k}$ is the ath element of V^*, which has k elements (i.e., $a \in \{1, 2, \ldots, k\}$). If the first element in V^* represents a prioritized vertex then the SOM selection is simply this element:

$$\mathbf{v_{i,j}} = \mathbf{v}_1^{*k} \quad \text{if } f \neq \emptyset \quad \text{in } n_j^f \tag{12}$$

If there are no prioritized vertices then a separate decision process is implemented.

The set E represents a number line that linearly maps to V^*:

$$E \rightarrow V^* \tag{13}$$

such that

$$E_0 \rightarrow \mathbf{v}_1^{*k} \tag{14}$$

and

$$E_3 \rightarrow \mathbf{v}_k^{*k} \tag{15}$$

A random number r is generated from a normal distribution with a mean of zero and a variable standard deviation (σ). When $\sigma = 1$, there is a 99.7% chance that $|r|$ will lie between 0 and 3. The vertex is chosen such that

$$E_{|r|} \rightarrow \mathbf{v}_a^{*k} = \mathbf{v_{i,j}} \tag{16}$$

with

$$a = \left\lceil \frac{|r|}{3/k} \right\rceil \tag{17}$$

(i.e., a is rounded up to the nearest integer).

For example, suppose V^* contains 20 vertices (i.e., $k = 20$) if $|r| = 0.5$, then $a = 4$. The selected vertex is, therefore, the fourth vertex in V^*:

$$E_{|0.5|} \rightarrow \mathbf{v}_4^{*20} = \mathbf{v_{i,j}} \tag{18}$$

The complete selection process can be visualized more clearly in Fig. 5, where there is no vertex prioritization applied to the vertices. Equations (16) and (17) are invalid for cases where $|r| > 3$, thus a final constraint needs to be introduced:

$$\mathbf{v_{i,j}} = \max_{l_{i,j}} U_i \equiv \mathbf{v}_k^{*k}\colon |r| > 3 \tag{19}$$

Because $|r|$ is weighted toward zero, the selection of $\mathbf{v_{i,j}}$ is weighted toward those $\mathbf{v_{i,j}} \in V^*$ with shorter task completion times $l_{i,j}$. The strength of this weighting can be modified by altering σ. As $\sigma \rightarrow 0$, the probability of $|r|$ being a small number increases, thus the

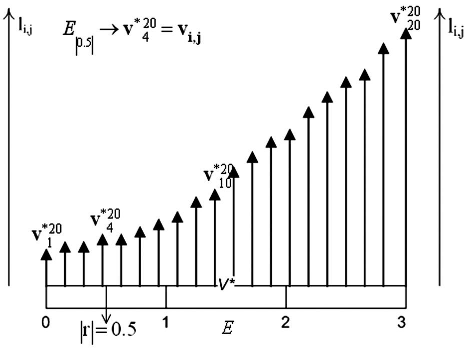

Fig. 5 Example vertex selection using the SOM.

Table 1 Task durations for stars of different spectral type [6]

Spectral Type			*F*	*G*	*K*	*M*
Task Time (days)	Detection		2.410	0.533	0.339	0.300
	Spectroscopy	<7.2 μm	1676 [a]	85.80	38.30	10.89
		8–9.2 μm	9.450	1.560	0.987	0.460
		9.2–10 μm	34.60	6.620	3.950	1.965
		10–13.2 μm	1.900	0.386	0.228	0.116
		13.2–17.2 μm	3.410	1.040	0.618	0.376
		>17.2 μm	62.50	34.90	26.10	20.46

[a] The *F*-type star less than less than 7.2 μm spectroscopy task duration is 1676 days as defined in [6]. However [6] notes that this is too long and advises measures to decrease this time.

Table 2 Simulated task distribution for five mission stages

Taskflag		Year0	Year1	Year2	Year3	Year4
Detection tasks	Completed	0	1044	1313	1341	1341
	Remaining	1341	297	28	0	0
Planets	Found	0	38	47	47	47
	Not found	47	9	0	0	0
Spectroscopy tasks	Completed	0	60	160	201	222
	Remaining	282	222	122	81	60

Table 3 Science operations planning considerations and their implementation

Constraint	Inclusion	Implementation
Maneuver time	Partially	Fixed at 0.02° s^{-1}
Calibration time	No	N/A
Formation pointing constraints	Yes	Field of view moves at a fixed rate of 0.986°/day
Detection task scheduling	Partially	At least one other task must be scheduled before a detection task can be repeated [a]
Planet detection analysis time	No	Spectroscopy can be scheduled immediately after third detection task
Planet orbital characteristics	No	N/A
Science task weighting	No	Tasks are given equal weighting

[a] With this definition the time between repeated detection tasks will up to approximately 2.5 days (Table 1), however, in reality the required separation time is likely to be much larger (greater than 90 days) depending on the orbital characteristics of the planet

likelihood of selecting shorter vertices increases. Eventually, however, σ will be so small that only one choice of vertex becomes available, and the algorithm will emulate the benchmark. σ is a tunable parameter within this algorithm.

This stochastic approach to vertex selection is not locally optimal but seeks to find better tours than the benchmark as it has the chance to examine a larger area of the solution space. It is unlikely, however, that a better solution will be found on the first attempt, and so this approach is enhanced by multiple iterations with the chosen tour being the best performing of the family of tours generated. The number of iterations *I* is another tunable parameter for this algorithm and greatly affects its performance and efficiency.

V. Initial Conditions and Definitions

To analyze the performance of the BP and SOM planning algorithms it is necessary to define some initial conditions that affect the calculations.

A. Star Catalogue and Observation Tasks

The star catalogue used for the algorithm contains coordinate data for the stars and task duration data for the observations to be performed on those stars. The list of target stars for the Darwin mission contains 447 stars [6]. Star distribution is fairly even in both ecliptic longitude and latitude, which assists the planning algorithms in avoiding large and time-consuming retarget maneuvers. For each star there are 7 observation tasks that can be performed, one planet detection task, and six individual spectroscopy tasks (defined by the spectral band observed). The task duration for every task is set depending on the spectral class of the star as in Table 1. Note how the duration of the detection tasks is generally less than the duration of the spectroscopy tasks. As there are 7 tasks for each of the 447 stars, the star catalogue holds 3129 task duration values. However, because each detection task is required to be repeated 3 times [6] the total number of potential tasks for the mission is 4023.

B. Simulating Planet Detection

In order for the algorithms to allow the scheduling of spectroscopy tasks some of the stars in the catalogue must have planets. As the existence of these planets is not known, it is necessary to simulate this requirement. The simulated probability of a star having a planet is arbitrarily set to approximately 10% giving 47 stars from the catalogue chosen randomly to host one planet each with the distribution even in ecliptic longitude but slightly biased in negative ecliptic latitude. It is reasonable to assume that no one portion of the sky will yield more planets than the next, and so this distribution can be viewed as fairly typical of what one might

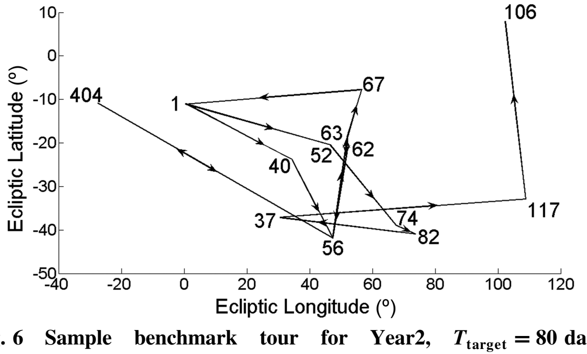

Fig. 6 Sample benchmark tour for Year2, $T_{\text{target}} = 80$ days, Startstar = 1.

Table 4 Sample benchmark tour (Year2, $T_{\text{target}} = 80$ days, Startstar = 1)

Star	Task	Time, days
1	1	0
40	1	2.43
56	1	4.85
404	6	8.30
56	1	10.76
62	1	13.18
63	1	15.59
62	1	18.00
63	1	20.41
56	1	22.84
63	1	25.26
67	4	29.22
1	4	33.20
52	4	39.85
74	4	46.48
82	2	57.37
37	2	68.28
117	2	79.21
106	2	90.12

find in reality. As previously calculated there are 4023 potential tasks for the mission. However this number was calculated assuming a 100% planet-to-star probability. Using the 10% planet-to-star probability shows that, for this analysis, there are 1623 tasks for the mission of summed duration approximately 14.8 years (far longer than the planned 5-year mission lifetime).

C. Initializing Mission Stage Markers

The performance of both of the planning algorithms will be affected by the time in the mission the algorithms are run. At the beginning of the mission, with no planets detected, the majority of tasks scheduled will be the shorter detection tasks. As the mission progresses, less detection tasks will be available and the longer spectroscopy tasks will start to be scheduled (because planets will have been found). By the end of the 5-year mission duration only spectroscopy tasks will be scheduled, as all the detection tasks will have been performed. Performance analysis of the planning algorithms at different stages of the mission is carried out using taskflag files that record a snapshot of the current task distribution for every year of the mission as in Table 2. Here “Year0” describes the beginning of the mission, “Year1” describes the end of the first year of the mission, and so on. The data in Table 2 were obtained using the BP with a target tour time of 365 days, 730 days, etc. Table 2 shows by the beginning of the fourth year of the mission (“Year3” taskflag file) all the detection tasks have been completed and only spectroscopy tasks remain.

D. Operating Conditions

In Sec I a list of planning considerations was given specifically relating to science operations planning for the Darwin mission. The algorithms described in this paper do not take into account all the complexities involved with these considerations. Table 3 lists the planning considerations and how they are implemented within the software. The omission of some of the considerations is mainly due to lack of data and the desire to simplify the algorithm for onboard operation. The partial implementations are work-arounds due to lack of data but still add the same level of complexity to the problem.

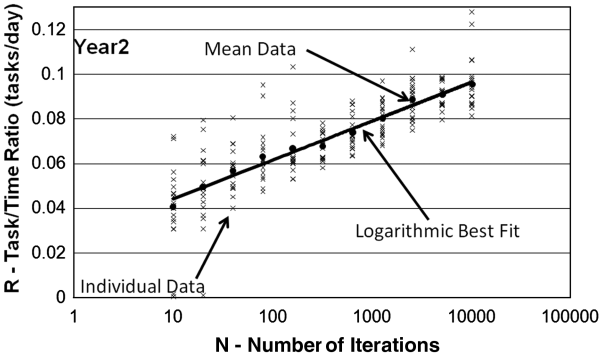

Fig. 7 R vs I for Year2, $T_{\text{target}} = 80$ days, Startstar = 1.

E. Hardware

The analysis was performed within the MATLAB 2006a software environment running in Windows XP Professional on a 3.06 GHz Intel Pentium 4 PC with 1 GB of RAM. It is assumed that these computer resources would be similar to those available onboard spacecraft by an estimated launch date circa 2030.

VI. Planning Algorithm Examples

This section shows examples of the tours optimized by the BP and SOM planning algorithms and includes details on how the variable parameters in the SOM were tuned.

A. Benchmark Planner Example

A sample BP tour, shown in Fig. 6, was generated using the Year2 taskflag file $T_{\text{target}} = 80$ days and starting from star 1 in the catalogue. The star index numbers from the mission catalogue can be seen next to the stars’ ecliptic coordinates. The stars are joined by a dashed line (--), and arrows (→) represent the direction of travel. For clarity the tour data are given in Table 4. The stars are represented by their index numbers in the mission catalogue. Task 1 represents a detection task and tasks 2, 4, and 6 represent the less than 7.2 μm, 9.2–10 μm, and 13.2–17.2 μm spectroscopy tasks, respectively, (as in Table 1). The time row represents the cumulative completion time of the task at each star (incorporating the maneuver time to retarget to the star). The entire tour takes 90.12 days to complete and with 18 tasks gives $R = 0.1997$ tasks/day.

The tour is very dynamic with large retarget maneuvers implemented both in increasing and decreasing ecliptic longitude. This is because the field of view moves very slowly in relation to the frequency of the tasks scheduled and the maneuver time is significantly less than the observation time. There are a number of parts of the tour that involve multiple retargets between two stars performing detection tasks (i.e., between stars 62 and 63 and between stars 63 and 56). This is due to the constraint that prevents detection tasks being scheduled consecutively. This implementation of the detection task scheduling constraint (Table 3) creates a scheduling situation that is unlikely to be replicated on a real mission, as the detection task repetition rate would have to be far lower than that simulated here.

Performance data for tours generated using the BP at other stages in the mission are shown in Table 5. For this data all the tours start at the same place, star 1 in the mission catalogue. The difference in T_{target} for the different mission stages is partly to allow for long enough tours to be generated (because near the end of the mission only the longer spectroscopy tasks will remain to be scheduled) and

Table 5 Performance data for the BP algorithm

Taskflag	Year0	Year1	Year2	Year3	Year4
T_{target}, days	5	40	80	160	320
R, tasks/days	3.298	1.553	0.200	0.051	0.039
Mean task duration, days	0.303	2.386	5.301	10.384	25.397
p	17	63	18	9	17
p_{waits}	0	0	0	0	4

Table 6 SOM tour performance extrapolation

Taskflag	Logarithmic fit equation	Benchmark R, tasks/day	Iterations required to reach benchmark	Mean iteration duration, s	Time to reach benchmark, h
Year0	$y = 0.0154\,\ln(x) + 3.0853$	3.2977	$\sim 9.76 \times 10^5$	0.623	~169
Year1	$y = 0.034\,\ln(x) + 0.0858$	1.5533	$\sim 5.55 \times 10^{18}$	0.064	$\sim 9.86 \times 10^{13}$
Year2	$y = 0.0076\,\ln(x) + 0.0269$	0.1997	$\sim 7.49 \times 10^9$	0.048	~99866
Year3	$y = 0.0018\,\ln(x) + 0.0271$	0.0510	$\sim 5.84 \times 10^5$	0.042	~6.81
Year4	$y = 0.0027\,\ln(x) + 0.0078$	0.0389	$\sim 1.00 \times 10^5$	0.131	~3.64

partly to ease the computation burden on the stochastic SOM (a large T_{target} for Year0 takes a long time to calculate due to the number of available tasks and the short duration of those tasks). The data show an apparent performance decrease as the mission progresses. This is to be expected as the availability of shorter tasks decreases as the mission progresses resulting in an increase in the mean task duration.

B. Tuning the Science Operations Module

The SOM has two tunable parameters that affect its performance and its computational burden. The first is the standard deviation of the random number generator σ used in the vertex selection process. A high σ increases the likelihood a longer duration vertex will be chosen whereas a small σ decreases that likelihood. However, σ cannot be too small or else no choice is available and the algorithm emulates the BP. The second parameter is the number of iterations I that are performed before the tour is selected. The time required for each iteration to complete depends upon T_{target} and the mission stage selected. For example, the longest iteration duration would be for a high T_{target} at stage Year0. Better performing tours are found using more iterations but at a cost of calculation time. Tuning σ and I is important to achieve the best balance between performance and time efficiency for the SOM.

1. Tuning the Number of Iterations I

The first parameter to tune is the number of iterations I. Figure 7 shows the relationship between the number of iterations and the best tour performance. The data were generated for mission stage Year2, $T_{\text{target}} = 80$ days, and starting at star 1 in the mission catalogue and $\sigma = 1$. The data points (×) represent the 20 generated tours for each I, and the data points (●) represent the mean. The line represents a logarithmic best fit of the mean. As expected, tour performance increases with increasing I. However, because of the logarithmic nature of the progression, the differences in R at high numbers of iterations become very small. Because the calculation time associated with I follows a linear trend, the SOM appears to operate within the law of diminishing returns. Doubling the number of iterations at high I only marginally increases the tour performance. This trend is also seen for the other mission stages and leads to a trade-off between maximizing I for tour performance and minimizing I for calculation efficiency. Comparing the tour performance for 10,000 iterations in Fig. 7 and the benchmark tour cost from Table 5 for the Year2 mission stage, it is clear to see that the mean SOM performance is approximately 50% less than the BP. By increasing I for the calculation the SOM should eventually be able to match the BP's tour performance, and further increasing I will eventually lead the SOM to the globally optimal tour for the initial conditions.

Table 6 shows the length of calculation time required by the SOM to reach the benchmark performance values from Table 5. Assuming the logarithmic best fit of the mean data from the R vs I data continue to be valid as I increases, the best fit equations are used to extrapolate the number of iterations required to reach the benchmark R. Mean iteration duration data for each mission stage are then used to calculate the time required to obtain performance results similar to the BP. The calculation times in Table 6 show that the calculation time required by the SOM to attain the benchmark performance ratios varies from approximately 3.6 h to an unattainable 11 million years. Table 6, therefore, shows that increasing I is generally not a feasible method to increase the performance of the SOM and that tuning σ from a value of $\sigma = 1$ is the only other alternative. This tuning, however, does require a fixed value for I.

Because the SOM is designed to run during the observation phase of the Darwin mission there is a limit to how long the calculations can take. In Table 1 the 10–13.2 μm spectroscopy task for M-type stars is the shortest observation at 0.116 days (approximately 10,000 s), therefore the SOM cannot not take any longer than this to complete its search for a tour. The most appropriate limit, therefore, is to impose a 10,000 s calculation time limit rather than a number of iterations limit. This calculation time limit was used for all further SOM calculations in this paper.

2. Tuning the Standard Deviation, σ

From the I-tuning analysis it is shown that when $\sigma = 1$ the SOM is unable to find tours performing better than the benchmark without violating the calculation time limit. This is because the size of σ determines the size of the solution space that algorithm has to search though. If the solution space is too large then the algorithm cannot find the "good" tours due to all the "bad" ones also present. Reducing the size of σ decreases the size of the solution space making it less likely for bad tours to be generated. σ however cannot be reduced indefinitely as there is a minimum boundary σ-value that causes the SOM to emulate the BP. This can be seen in Fig. 8. The chart shows mean tour performance for varying σ for the Year1 taskflag with $T_{\text{target}} = 40$ days. The standard deviation is plotted on a logarithmic scale whereas R is plotted on a linear scale. The data points (●) are the mean performance from 20 repetitions of the algorithm. The individual repetition data are not shown as the spread is indiscernible from the mean data points. Also shown is the benchmark performance for Year1 at 1.5533 tasks/day represented by a dashed line.

The data in Fig. 8 show that the performance increases with decreasing σ surpassing the benchmark performance between $\sigma = 0.1$ and 0.2. Decreasing σ past this point increases the performance to a maximum ($\sigma = 0.07$) before it falls again to the benchmark value. This shows that with careful setting of σ the SOM is able to find better performing tours than the BP within the 10,000 s calculation time limit. These results are the same for the other mission stages except the σ value that generates the highest performing tour changes.

Table 7 shows how the optimal σ changes at different mission stages by decreasing from 0.1 to 0.03 throughout the mission. This is due to task availability at each stage of the mission. Because of the large number of available tasks early on in the mission, the long duration tasks are very unlikely to be selected and so a relatively large σ can be maintained. Later on in the mission, however, task availability is greatly reduced, and so the likelihood of selecting a

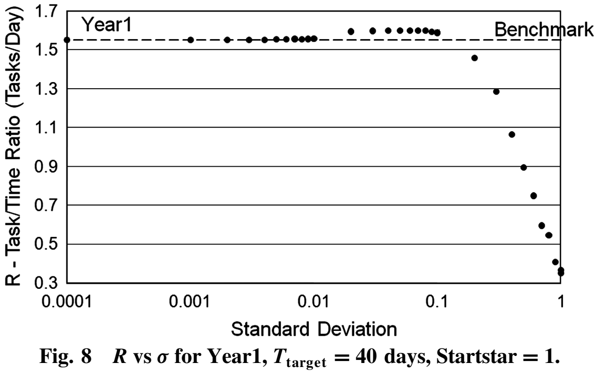

Fig. 8 *R* **vs** *σ* **for Year1,** $T_{\text{target}} = 40$ **days, Startstar = 1.**

Table 7 Change of optimal σ with mission stage

	T_{target}, days	Optimal σ	R_{BP}, tasks/day	Mean R_{SOM}, tasks/day
Year0	5	0.1	3.2976	3.6241
Year1	40	0.07	1.5533	1.5996
Year2	80	0.07	0.1997	0.2204
Year3	160	0.04	0.0509	0.0557
Year4	320	0.03	0.038904	0.038905

long duration task is greatly increased. Reducing σ lowers the chance of a long duration task being chosen. If the same σ is maintained throughout the mission then tour performance will suffer in the later stages. If σ is allowed to decrease, however, maximum tour performance can be maintained.

C. Science Operations Module Example

A sample SOM tour is shown in Fig. 9 generated using the Year2 taskflag file, $T_{\text{target}} = 80$ days, and starting from star 1 in the catalogue (the same initial conditions as the example BP tour, Table 4 and Fig. 6). The star index numbers from the mission catalogue can be seen next to the star's ecliptic coordinates. The stars are joined by a dashed line (--) and arrows (→) represent the direction of travel. For clarity the tour data are given in Table 8. The stars are represented by their index numbers in the mission catalogue. Task 1 represents a detection task and tasks 2, 3, 4, and 6 represent the less than 7.2 μm, 8–9.2 μm, 9.2–10 μm, and 13.2–17.2 μm spectroscopy tasks, respectively. The time row represents the cumulative completion time of the task at each star (incorporating the maneuver time to retarget to the star). The entire tour takes 81.62 days to complete and with 18 tasks gives $R = 0.2206$ tasks/day. Although the BP example tour also scheduled 18 tasks it took an extra 8.5 days to complete; therefore, the SOM example is a much better performing tour.

Table 8 Sample SOM tour (Year2, $T_{\text{target}} = 80$ days, Startstar = 1)

Star	Task	Time, days
1	1	0
400	4	3.97
404	6	7.38
404	3	16.83
1	4	20.80
56	1	23.24
40	1	25.66
62	1	28.08
63	1	30.49
62	1	32.91
67	4	36.86
52	4	43.49
63	1	45.91
56	1	48.33
37	2	59.23
74	4	65.87
56	1	68.28
63	1	70.71
82	2	81.62

VII. Planning Algorithm Comparison

In this section the SOM and the BP algorithms are compared to assess their abilities in solving the science task assignment problem. A comparison of the data from the BP analysis and the SOM analysis can be found in Table 9. For each mission stage the mean task/time ratio (R_{SOM}) for the SOM algorithm from Table 7 is compared with the BP algorithm results (R_{BP}) from Table 5. The performance increase is a measure of how much larger R_{SOM} is than R_{BP} as a percentage of R_{BP}. The difference between R_{SOM} and R_{BP} and the mean task duration D (taken from Table 5) is used to calculate the mean rate of extra observation time gained (hours/day) during each mission stage if using the SOM instead of the BP to plan the schedule:

$$\dot{O} = 24 \times (R_{\text{SOM}} - R_{\text{BP}})D \tag{20}$$

The data in Table 9 show that by tuning the number of iterations (or fixing the calculation time) and the standard deviation σ the SOM is able to outperform BP in task scheduling. The increase in performance varies for different mission stages. This is partly due to the different task availability and task characteristics at each mission stage and partly due to the starting star of the tour. The Year4 data from Table 9 show no significant increase in tour performance using the SOM over the BP. This is due to a combination of the starting star of the tour (described in the following) and the fact that near the end of the mission there is simply little choice available in scheduling the tours (only 60 tasks remain for Year4, Table 2). With this lack of choice and the fact that the remaining tasks will all have similar duration times increases the likelihood that the BP and SOM tours will have similar performances.

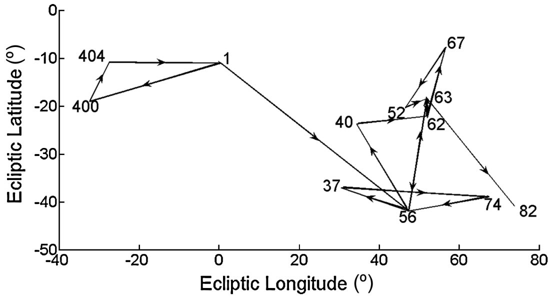

Fig. 9 Sample SOM Tour for Year2, $T_{\text{target}} = 80$ days, Startstar = 1.

Another factor affecting the performance variations is the starting star for the tour. The SOM will produce very similar performing tours for starting star indexes adjacent to one another in the mission catalogue because the task availability for these starting stars will be roughly the same. For starting stars separated my many degrees of ecliptic longitude, however, tour performance (and comparison with the BP) will vary noticeably. Table 10 shows comparison data generated exactly the same way as for Table 9 but with the starting star at star 336 in the mission catalogue. The ecliptic longitude of star 336 is approximately −90° from star 1. The performance increases in Table 10 are much lower than in Table 9 emphasizing the effect of the tour starting star on the comparative performance of the SOM. The data show, however, that the SOM has a performance increase over the BP over all mission stages, and comparing other tours generated with different starting stars confirms this. The SOM, therefore, is able to consistently outperform the BP (with varying degrees of performance) for all mission stages and all tour starting stars.

In real terms the performance increase can be described as the increased observation rate for a schedule optimized by the SOM. The last rows in Tables 9 and 10 show this rate using the mean task duration from Table 5. The data show that up to 3 hours/day extra can be assigned for science observations when schedule planning using the SOM. This demonstrates the SOM's ability to generate tours that are more time efficient, maximizing observation time and the number of observations performed, hence minimizing the time taken to perform the reconfiguration maneuvers to move the formation from one task to another.

Table 9 Comparison data between the SOM and BP algorithms using Startstar = 1

Mission stage, taskflag	Year0	Year1	Year2	Year3	Year4
Target tour time T_{target}, days	5	40	80	160	320
Optimal standard deviation, σ	0.1	0.07	0.07	0.04	0.03
Mean SOM ratio R_{SOM}, tasks/day	3.6241	1.5996	0.2204	0.0557	0.038905
BP ratio R_{BP}, tasks/day	3.2976	1.5533	0.1992	0.0509	0.038904
Performance increase of SOM, %	9.9	3.0	11.6	9.4	0.003
Ratio difference, tasks/day	0.3265	0.0463	0.0232	0.0048	1×10^{-6}
Mean task duration$D$, days	0.3032	2.3858	5.3014	10.383	25.3966
Mean extra observation rate $\dot{O}$, h/day	~2.4	~2.7	~3.0	~1.2	~0

Table 10 Comparison data between the SOM and BP algorithms using Startstar = 336

Mission stage, taskflag	Year0	Year1	Year2	Year3	Year4
Target tour time T_{target}, days	5	40	80	160	320
Optimal standard deviation, σ	0.1	0.07	0.07	0.04	0.03
Mean SOM ratio R_{SOM}, tasks/day	3.3997	1.7084	0.2357	0.0601	0.039381
BP ratio R_{BP}, tasks/day	3.2868	1.7065	0.2220	0.0600	0.036872
Performance increase, %	3.4	0.11	6.2	0.2	6.8
Ratio difference, tasks/day	0.1129	0.0019	0.0137	0.0001	0.0025
Mean task duration D, days	0.3032	2.3858	5.3014	10.383	25.3966
Mean extra observation rate $\dot{O}$, h/day	~0.8	~0.1	~1.8	~0.025	~1.5

VIII. Further Development

A number of the complex timing constraints introduced in Sec. I have not been implemented in this version of the SOM (Table 3). This has partly to do with a lack of information regarding these constraints and partly a desire to simplify the problem to give the algorithm a performance that would enable its operation onboard a spacecraft. The constraints not modeled include calibration time, detection task scheduling, planet detection analysis time, and the planetary orbital characteristics. Constraints partially modeled or not implemented include the maneuver time and the science task weighing. Fully implementing these constraints would provide a much more accurate tour reflective of the Darwin mission.

1) Detection task scheduling and the planetary orbital characteristics could be modeled by giving each of the tasks a time range in which they can be executed. This range would be calculated for each task to ensure the correct order of tasks performed (i.e., detection before spectroscopy), appropriate task separation (i.e., between detection tasks), and correct planetary alignment (i.e., to avoid occultation or transits during observations). The time range would be referenced during the core algorithm (Sec. IV.B, the reduction of V_i to U_i) and only tasks for which time range included t_i would be selected for U_i. Should a time range pass and the task not completed, then the time range would need to be updated for the next appropriate range.

2) The calibration time constraint could be implemented by adding this duration to Eq. (1).

3) Planet detection analysis time could be implemented in two ways. In one implementation spectroscopy tasks are simply unschedulable (through the use of a flag) until detection has been confirmed. This is simple to implement but will give less accurate schedules due to the unknown analysis duration. In another implementation the time range concept introduced previously can be used with the time range offset to account for a standard or expected analysis time.

4) A more accurate maneuver time can be modeled by including a maneuver planning algorithm (such as the PAM [25,26] or the TDM [26] introduced in Sec. III). The inclusion of this algorithm would not only give more accurate maneuver durations but also allow for the analysis of other concepts, like fuel consumption. The disadvantage, however, would be the increased complexity of the algorithm decreasing the number of iterations that can be achieved within any given time limit and, therefore, affecting the algorithm's performance.

5) Finally, the science task weighting can be modeled as suggested in the problem formulation (though not implemented in the analysis).

IX. Conclusions

The science operations module (SOM) and the benchmark planner (BP) are two algorithms that aim to solve the science task assignment (STA) problem for separated spacecraft interferometry missions like Darwin. This paper has introduced the SOM as part of a larger maneuver planning architecture (MPA) that aims to solve the optimal spacecraft formation flying maneuver planning problem at all levels including task scheduling, station keeping, and formation reconfiguration. Analysis of the SOM shows that with careful tuning of its two variable parameters it is capable of generating science operations schedules that outperform the simple BP (when the performance metric is to maximize the ratio between the number of observations scheduled and the time taken to complete the schedule). Though the performance increase varies greatly depending on the starting star for the schedule and the mission stage, the SOM is able to increase the amount of time spent performing science observations by up to 3 hours/day. This emphasizes the necessity for science operations planning for separated spacecraft interferometry missions when the total number of possible observations far outweighs the mission duration.

Acknowledgments

This work was carried out in a joint collaboration between Cranfield University, the European Space Agency, and EADS Astrium. The authors wish to extend their thanks to Stephen Kemble, EADS Astrium, for his valuable contributions.

References

[1] Blackwood, G. H., Lay, O. P., Deininger, W. D., Gudim, M. A., Ahmed, A., Duren, R. M., Noecker, C., and Barden, B., "StarLight Mission: A Formation Flying Stellar Interferometer," *Proceedings of the SPIE—Interferometry in Space*, Vol. 4852, 2003, pp. 463–480. doi:10.1117/12.460942

[2] Beichman, C., Gomez, G., Lo, M., Masdemont, J., and Romans, L., "Searching for Life with the Terrestrial Planet Finder: Lagrange Point Options for a Formation Flying Interferometer," *Advances in Space Research*, Vol. 34, No. 3, 2004, pp. 637–644. doi:10.1016/j.asr.2003.05.032

[3] Folta, D., Hartman, K., Howell, K., and Marchand, B., "Formation Control of the MAXIM L2 Libration Orbit Mission," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA, Reston, VA, Aug. 2004, pp. 997–1014.

[4] Tinto, M., "LISA: The Laser Interferometer Space Antenna," NASA JPL TRS 1992+, Dec. 2003.

[5] Kaltenegger, L., Karlsson, A., Fridlund, M., and Absil, O., "Overview of the Darwin Mission," *Towards Other Earths: DARWIN/TPF and the Search for Extrasolar Terrestrial Planets*, ESA-SP-539, ESA, Noordwijk, The Netherlands, 2003, pp. 459–464.

[6] Karlsson, A., Kaltenegger, L., den Hartog, R., d'Arcio, L., Kilter, M., Erd, C., and Ankersen, F., "Darwin, TTN+ Mission Design Assessment," ESA, Report SCI-A/2004/187/Darwin/DMS, Noordwijk, The Netherlands, Nov. 2004.

[7] Muscettola, N., Pell, B., Hansson, O., and Mohan, S., "Automating Mission Scheduling for Space-Based Observatories," *Robotic Telescopes: Current Capabilities, Present Developments, and Future Prospects for Automated Astronomy*, edited by G. W. Henry and J. A. Eaton, Vol. 79, Astronomical Society of the Pacific, Provo, UT, 1995, pp. 148–166.

[8] Ferri, P., and Sorensen, E., "Automated Mission Operations for Rosetta," *Proceedings of Space Ops '98: 5th International Symposium on Space Mission Operations and Ground Data Systems* [online], Paper 1b014, June 1998, http://track.sfo.jaxa.jp/spaceops98/paper98/track1/1b014.pdf [retrieved 29 Jan. 2009].

[9] Smith, B., Rajan, K., and Muscettola, N., "Knowledge Acquisition for the Onboard Planner of an Autonomous Spacecraft," *Lecture Notes in Computer Science*, Vol. 1319, 1997, pp. 253–268. doi:10.1007/BFb0026790

[10] Chein, S., Smith, B., Rabideau, G., Muscettola, N., and Rajan, K., "Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft," *IEEE Intelligent Systems and Their Applications*, Vol. 13, No. 5, 1998, pp. 50–55. doi:10.1109/5254.722362

[11] Rui, X., Ping-Yuan, C., Xiao-Fei, X., and Hu-Tao, C., "Design for Autonomous Mission Planning System," *Aircraft Engineering and Aerospace Technology*, Vol. 75, No. 4, 2003, pp. 365–371.

[12] Rui, X., Ping-Yuan, C., and Xiao-Fei, X., "Realization of Multi-Agent Planning System for Autonomous Spacecraft," *Advances in Engineering Software*, Vol. 36, No. 4, 2005, pp. 266–272. doi:10.1016/j.advengsoft.2004.10.003

[13] Bailey, C. A., McLain, T. W., and Beard, R. W., "Fuel-Saving Strategies for Dual Spacecraft Interferometry Missions," *Journal of the Astronautical Sciences*, Vol. 49, No. 3, 2001, pp. 469–488.

[14] Beard, R. W., and Hadaegh, F. Y., "Fuel Optimized Rotation for Satellite Formations in Free Space," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, Vol. 5, 1999, pp. 2975–2979. doi:10.1109/ACC.1999.782306

[15] Bansal, N., Blim, A., Chawla, S., and Meyerson, A., "Approximation Algorithms for Deadline TSP and Vehicle Routing with Time Windows," *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 2004, pp. 166–174.

[16] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, Cambridge, England, UK, 1992, Chap. 10, p. 444.

[17] Hernádvölgyi, I. T., "Solving the Sequential Ordering Problem with Automatically Generated Lower Bounds," *Proceedings of Operations Research*, Springer–Verlag, New York, 2003, pp. 355–362.

[18] Hurkens, C. A., and Woeginger, G. J., "On the Nearest Neighbor Rule for the Traveling Salesman Problem," *Operations Research Letters*, Vol. 32, No. 1, 2004, pp. 1–4. doi:10.1016/S0167-6377(03)00093-2

[19] Merz, P., and Freisleben, B., "Genetic Local Search for the TSP: New Results," *Proceedings of the IEEE International Conference on Evolutionary Computation*, IEEE Publications, Piscataway, NJ, 1997, pp. 159–164. doi:10.1109/ICEC.1997.592288

[20] Dorigo, M., Di Caro, G., and Gambardella, L. M., "Ant Algorithms for Discrete Optimization," *Artificial life*, Vol. 5, No. 2, 1999, pp. 137–172. doi:10.1162/106454699568728

[21] Brucker, P., Hilbig, T., and Hurink, J., "A Branch and Bound Algorithm for a Single-Machine Scheduling Problem with Positive and Negative Time Lags," *Discrete Applied Mathematics*, Vol. 94, Nos. 1-3, 1999, pp. 77–99. doi:10.1016/S0166-218X(99)00015-3

[22] Crauwels, H. A. J., Potts, C. N., and Van Wassenhove, L. N., "Local Search Heuristics for the Single Machine Total Weighted Tardiness Scheduling Problem," *INFORMS Journal of Computing*, Vol. 10, No. 3, 1998, pp. 341–350. doi:10.1287/ijoc.10.3.341

[23] Franca, P. M., Mendes, A., and Moscato, P., "A Memetic Algorithm for the Total Tardiness Single Machine Scheduling Problem," *European Journal of Operational Research*, Vol. 132, No. 1, 2001, pp. 224–242. doi:10.1016/S0377-2217(00)00140-5

[24] Burgon, R., Roberts, J. A., and Roberts, P. C. E., "Optimal Path Planning for Spacecraft Formation Flying: Planning Architecture and Operations," *Advances in the Astronautical Sciences*, Vol. 129, 2007, pp. 2685–2704.

[25] Burgon, R., Roberts, P. C. E., and Ankersen, F., "Optimal Autonomous Manoeuvre Planning for Spacecraft Formation Flying—Position Assignment," *Advances in the Astronautical Sciences*, Vol. 130, 2008, pp. 1015–1032.

[26] Burgon, R., Roberts, P. C. E., and Ankersen, F., "Two-Stage Optimal Manoeuvre Planning For Spacecraft Formation Flying Missions," *Proceedings of the 3rd International Symposium on Formation Flying, Missions and Technologies*, [CD-ROM], ESA-SP-654, ESA, Noordwijk, The Netherlands, 2008.

C. McLaughlin
Associate Editor