Komninou, Eirini and Vasile, Massimiliano and Minisci, Edmondo (2012) *Optimal dynamic operations scheduling for small-scale satellites.* In: 63rd International Astronautical Congress, 2012-10-01 - 2012-10-05, Naples.

IAC-12-B4.3.7

OPTIMAL DYNAMIC OPERATIONS SCHEDULING FOR SMALL-SCALE SATELLITES

Eirini Komninou
University of Strathclyde, Glasgow, United Kingdom
eirini.komninou@strath.ac.uk

Massimiliano Vasile
University of Strathclyde, Glasgow, United Kingdom
massimiliano.vasile@strath.ac.uk

Edmondo Minisci
University of Strathclyde, Glasgow, United Kingdom
edmondo.minisci@strath.ac.uk

Abstract A satellite's operations schedule is crafted based on each subsystem/payload operational needs, while taking into account the available resources on-board. A number of operating modes are carefully designed, each one with a different operations plan that can serve emergency cases, reduced functionality cases, the nominal case, the end of mission case and so on. During the mission span, should any operations planning amendments arise, a new schedule needs to be manually developed and uplinked to the satellite during a communications' window. The current operations planning techniques offer a reduced number of solutions while approaching operations scheduling in a rigid manner. Given the complexity of a satellite as a system as well as the numerous restrictions and uncertainties imposed by both environmental and technical parameters, optimising the operations scheduling in an automated fashion can offer a flexible approach while enhancing the mission robustness. In this paper we present Opt-OS (Optimised Operations Scheduler), a tool loosely based on the Ant Colony System algorithm, which can solve the Dynamic Operations Scheduling Problem (DOSP). The DOSP is treated as a single-objective multiple constraint discrete optimisation problem, where the objective is to maximise the useful operation time per subsystem onboard while respecting a set of constraints such as the feasible operation timeslot per payload or maintaining the power consumption below a specific threshold. Given basic mission inputs such as the Keplerian elements of the satellite's orbit, its launch date as well as the individual subsystems' power consumption and useful operation periods, Opt-OS outputs the optimal ON/OFF state per subsystem per orbital time step, keeping each subsystem's useful operation time to a maximum while ensuring that constraints such as the power availability threshold are never violated. Opt-OS can provide the flexibility needed for designing an optimal operations schedule on the spot throughout any mission phase as well as the ability to automatically schedule operations in case of emergency. Furthermore, Opt-OS can be used in conjunction with multi-objective optimisation tools for performing full system optimisation. Based on the optimal operations schedule, subsystem design parameters are being optimised in order to achieve the maximal usage of the satellite while keeping its mass minimal.

Keywords – Optimal operations scheduling, Optimal satellite design, Ant Colony Optimisation, Multi-objective optimisation, Maximal Satellite Usage.

NOMENCLATURE

ρ/ξ	Global/Local Pheromone evaporation constant
τ_0	Initial pheromone level
τ_{ij}	Pheromone edge between node i and j
C	Mean power consumption throughout mission
α	Pheromone parameter weight
q_0	Exploration parameter
q	Randomly generated real number
p_{ij}	Probability of moving from node i to node j
k	Ant index with k = 1,2...m
m	Number of ants per colony
n	Number of timesteps of mission timeline
r	Number of revolutions
S	Number of subsystems onboard
OP_{max}	Maximum useful operation period
J_i^k	Candidate list of node i, ant k
Q^+	Best-so-far quality
I_{max}	Definite optimisation termination criterion
a	Semimajor axis
e	Eccentricity
i	Inclination
$RAAN$	Right Ascension of the Ascending Node
ω	Argument of periapsis
M_o	Mean anomaly at epoch

1 INTRODUCTION

Designing satellite operation plans is a strenuous task, deriving from the cooperative work of several disciplines. After examining the mission concept and supporting architecture, a preliminary mission plan is developed based on the mission objectives. According to the mission payloads and subsystems on-board, operation requirements are formed based on operation timing per payload or subsystem, link budgets, data budgets and so on.

Depending on the mission phase and information available, the operations plan is initially formed and then being updated whenever new information is arising. Occasionally information is not available since the operations design isn't very far along or not specified yet, thus leading to assumptions. When more information becomes available, the cost of changes can be determined by modifying the operations concept and re-evaluating the cost and complexity of the mission. Examples of objectives and constraints shaping mission operations planning are: Maximisation of real-time contact and commanding versus on-board autonomy and data-storage, maximisation of the involvement of educational institutions using amateur university-run ground stations for instance, limiting the image budget to a specific number of images, using a specific tracking network, limiting the mission cost etc.

Based on the launch dates/windows, trajectory profile, mission phases and the activities required during each phase, a final set of observation and operation strategies is devised, based on the mission description and constraints. While operation strategies derive from mission objectives, sometimes they highly depend on the designer's background or experience. Therefore, an identification of whether the strategies used are mandatory, highly desirable or based on personal and organisational preferences, needs to be performed throughout the mission operations planning process.

Apart from nominal operations, designers need to take into account various mission scenarios corresponding to reduced functionality operations (e.g post-launch activation phase), emergency operations (e.g instrumentation malfunction) etc. Such scenarios can be tackled by designing a specific set of operational modes, developed to minimise the probability of mission failure by performing a set of predefined actions aiming at isolating and solving possible errors that can occur. Each mode contains a different set of actions based on its functionality, for example a safe mode is designed in order to tackle a potential malfunction. For instance if the power availability is greatly decreased due to an electrical malfunction, all non-vital subsystems including payloads are turned off in order to save energy and avoid a power surge while the telecommunications are reduced to an absolute minimum.

Based on the aforementioned facts, it is clear that mission operations planning is not only a challenging and time consuming task, but can also be rigid and prone to errors. More than 43% of failures may be attributed to human error [1, 2, 3, 4], therefore enhancing the robustness of mission design is of uttermost importance. The proposed work aims at tackling these issues by offering tools and methodologies for dynamically performing optimal operations scheduling on the spot while maximising the satellite usage.

Furthermore, an integrated system-operation approach is presented, aiming at designing a full optimal satellite system. Using a multi-objective agent based optimiser [5], a Pareto set of points is formed corresponding to optimal satellite designs, with each Pareto point representing a low lying optimiser. Every point included in the Pareto set represents a vector of values corresponding to optimal design parameters per subsystem on-board the satellite leading to an optimal system design.

The aforementioned integrated approach can lead to robust system designs developed in a timely manner and at a lower cost, thus making space accessible to a wider spectrum of institutions with lower budgets.

2 DYNAMIC OPERATIONS SCHEDULING

Operations scheduling can be a highly dynamic process. Depending on the mission phase and complexity, the satellite's health and other factors that can affect the progress of the mission, new operations schedules need to be designed on a tight time-frame in order to ensure the mission success.

A way to tackle the aforementioned challenge is to automate and optimise the operations scheduling process. Automating the operations scheduling process can both optimise the satellite usage, enhance the design's robustness, as well as alleviating the designer from strenuous tasks. In order to create an automated process able to optimise operations scheduling, a modular approach is utilised comprising nature-inspired optimisation techniques in combination with satellite subsystem and environment modelling. Initially a set of modules is used in order to design the orbital, environmental and technical elements comprising the mission, based on the mission requirements and constraints. Once all elements have been established, they are passed to an Ant Colony System based optimiser in search of the optimal operations schedule based on the requirements and constraints set. The final result is an optimal operations schedule developed in an automated manner, on the spot.

The following sections aim at giving a background of the Ant Colony System structure and operation, followed by a more detailed view of the proposed tools and methodologiesl for performing optimal dynamic operations scheduling.

2.1 Ant Colony System

The Ant Colony System (ACS) [6, 7, 8, 9, 10] was developed by Dorigo and Gambardella as an evolution of the first agent based algorithm based on ants' behaviour, namely the Ant System (AS), able to solve complicated combinatorial optimisation problems more efficiently. It was first used for solving NP-hard problems, like the Traveling Salesman Problem (TSP) [11] where the shortest Eucledian distance path needs to be found among a set of cities to be visited.

All ant-based algorithms utilise a common basic idea. Ant colonies explore the search space in a pseudo-random manner, searching for possible routes from the ant nest (starting point) to the food source (ending point) with the aid of a 'compass' (heuristic information), aiming at finding the shortest path possible between the start and end. Ants exchange information with each other via the means of pheromone deposition, a chemical used in real life ant colonies for leading ants towards good solutions [12]. The process is iterative, aiming at finding the best solution possible throughout each iteration and sharing that experience throughout the next iterations. At the end of each iteration, ants deposit pheromone throughout the best-so-far path as a means of synergy thus communicating their experience to the rest of the agents. This enhances the pheromone level at this path, thus making it more attractive for the ants to follow. Since pheromone is time dependent, it evaporates with time, therefore paths which do not receive pheromone enhancement become less attractive for the agents to follow. The ACS incrementally forms paths while searching for the optimal solution, using the steps described in Algorithm 1.

Algorithm 1 ACS algorithm

```
Define $m, I_{last}, \tau_0, q_0, \alpha, \beta$
Form candidate list $J_i \ \forall i \in [search\ space]$
while $I_{max}$ not met do
  Place all agents on start
  for k = 1:m do
    while not terminate do
      Calculate $\eta_{ij}$
      Decide next $j^k$ using Eq. (2)
      Perform local pheromone update on edge $ij$
      using Eq. (4)
    end while
  end for
  Calculate the quality of all tours $T^k$
  Find $T^+$
  Perform global pheromone update on $T^+$ using
  Eq. (5)
end while
```

First, the basic algorithm parameters are initialised. A uniform layer of pheromone is laid on the search space in order to keep the probability distribution deriving from Eq. (3) positive. Lack of initial pheromone would lead to a probability distribution of zeros, thus halting the ants and preventing them from exploring the search space. The ants (also known as agents) start exploring the search space in a pseudo-random fashion (their exploration is biased by the existence of pheromone in combination with one or multiple heuristics i.e values acting as a compass), incrementally constructing possible optimal routes. Once all agents of a single colony complete a full tour from the starting to the ending point, routes' lengths are measured, the shortest route, T^+, is enhanced with pheromone and all agents are placed back to the starting point in order to proceed to the next round of exploration. This iterative process continues until the optimisation termination criterion, I_{max}, is met. A step by step explanation of Algorithm 1 can be found below.

- Define the number of agents ,m, per colony, optimisation termination criterion ,I_{last}, exploration vs exploitation parameter ,q_0, pheromone parameter weight ,α, heuristic parameter weight ,β. The initial pheromone level of the search space ,τ_0, is calculated as follows

  $$\tau_0(t) = (n \cdot L_{nn})^{-1} \tag{1}$$

  with L_{nn} being the length of the nearest neighbour tour, performed using the nearest-neighbour classification algorithm [13] ,n the total number of nodes included in the search space.

- Construct the candidate list of every node in the search space ,J_i, with $i \in [search\ space]$. In order to speed up the exploration process, the ACS agents get to choose their next step based on nodes included in a candidate list i.e a set of nodes J_i^k in the vicinity of node i where ant k is situated at. Unvisited nodes contained in the candidate list are examined first, having priority over the rest of the nodes contained in the search space.

- While the optimisation termination criterion ,I_{max}, is not met
- Place all ants on the starting node.
- For all the ants ,m, comprising a colony
- While the ants have not yet reached the ending node
- Calculate the length heuristic for edge ij ,$\eta_{ij} = L_{ij}^{-1}$, with L_{ij} being the euclidean length of edge ij.
- Choose the next ant step ,j, on the basis of the *pseudo-random proportional rule.* The transition of ant k, situated on node i, to node j, is decided according to Eq. (2)

$$j = \begin{cases} \arg\max_{u \in J_i^k} \{[\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \quad (2)$$

where q is a randomly generated real number uniformly distributed over $[0,1]$, q_0 is a tunable parameter in the interval $(0,1)$, $J \in J_i^k$ selected according to the transition probability distribution below

$$p_{ij}^k(t) = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta} \quad (3)$$

where τ_{ij} is the pheromone level of edge ij, α and β are weight parameters. Likewise, τ_{il} is the pheromone level of edge il and η_{il} is the heuristic value of edge il with l being every node comprising J_i^k. Tuning q_0 enhances the exploitation vs exploration option, where $q \leq q_0$ aims in exploiting the available knowledge whereas $q > q_0$ aims at exploring the available search space.

- Perform local pheromone update on edge ij. While exploring the search space, ants perform a local pheromone update after each step. Once ant k takes a step from node i to node j, it updates the pheromone level of edge ij according to Eq. (4).

$$\tau_{ij}(t) \leftarrow (1-\xi) \cdot \tau_{ij}(t) + \xi \cdot \tau_0(t) \quad (4)$$

where τ_{ij} is the pheromone level of edge ij, ξ is the local pheromone evaporation rate parameter, τ_0 is the initial pheromone trail value.

The *local pheromone update* aims at subtracting a small fraction of the pheromone on the edge that ant k just visited. That way, edge ij becomes less attractive to the ants to follow, thus enhancing exploration. By shuffling ants' paths with the use of the local pheromone update, the probability of finding better solutions and avoiding stagnation is increased.

- End the while condition
- End the for loop.
- Calculate the quality of each ant's tour ,T^k, on the basis of euclidean distance.
- Find the best-so-far tour ,T^+, corresponding to the shortest length tour.
- Perform a global pheromone update on the edges comprising the best-so-far tour ,T^+, according to the *global pheromone update rule*

$$\tau_{ij}(t) \leftarrow (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) \quad (5)$$

where ij are the edges comprising T^+, ρ is the pheromone evaporation rate parameter and

$$\Delta\tau ij(t) = (L^+)^{-1}$$

with L^+ being the length of T^+ The *global pheromone update* allows colony members to spread all acquired experience among them, promoting synergy between ants thus leading to a faster convergence towards the optimal solution.

- End the while condition.

The following section describes how the ACS philosophy is utilised in order to achieve optimal mission operations scheduling, showing that Ant Colony Optimisation algorithms can be applied to solving all types of problems, physical or conceptual ones.

2.2 Optimal Operations scheduler (Opt-OS)

Opt-OS is a tool designed for deriving the optimal mission operations schedule automatically, based on the mission requirements and available resources. The objective is to maximise the satellite usage by maximising the useful operation period per subsystem while respecting a set of constraints such as the feasible operation timeslot per payload or maintaining the maximum power consumption below a specific threshold.
It has a modular structure, with each module outputting a specific set of information used for shaping the resource requirements and availability on-board.

First, the orbit is being designed based on its classical orbital elements [14]. Based on the orbit, an initial operations plan, OP_{max}, is derived, corresponding to the maximum useful operation time period allocation per subsystem on-board.

The satellite is modeled using a set of analytical models able to estimate the physical and electrical characteristics per subsystem, based on a specific set of design and environmental inputs. Utilising the physical and electrical characteristics per subsystem as well as the OP_{max}, a search space is created corresponding to all the possible states that could be applied in order to derive an operations schedule. A nature-inspired metaheuristic, namely an Ant Colony System inspired algorithm, is then used for exploring the search space in search of the optimal operations schedule. The aforementioned process offers design flexibility and enhanced robustness as it allows the designer to derive an optimal operations schedule on the spot, respecting the constraints set such as the power availability, target visibility and so on. Furthermore, Opt-OS can be used in case of emergency or partial failure (e.g partial instrument or subsystem failure imposing unscheduled changes in the operations plan).

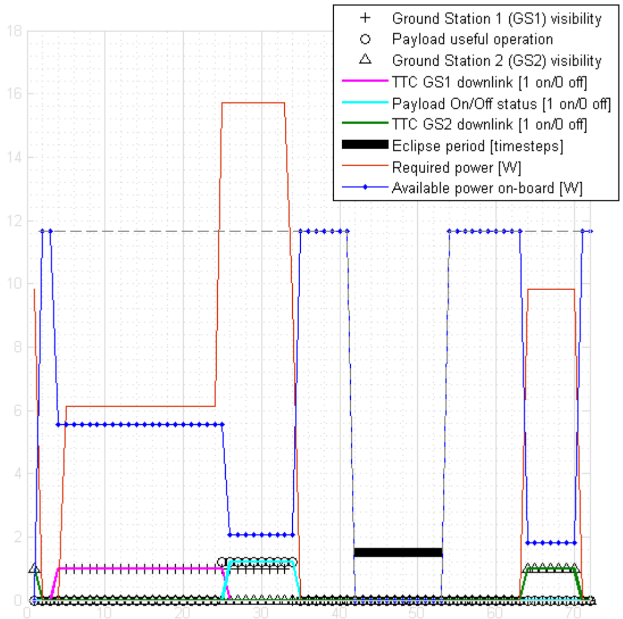


Figure 1: Optimal operations plan. Note that there is no available power during eclipse time (battery malfunction). Also, the generated power during daylight is limited.

Figure 1 shows an example of how Opt-OS can schedule operations on the spot based on the available resources on-board. In this instance the satellite's batteries cannot be charged due to a malfunction. Furthermore, the solar arrays are not generating enough power to support multiple subsystems operating simultaneously. Based on the power availability, Opt-OS schedules each subsystem's operation time automatically. The subsystem operation priority can be defined based on the importance of each subsystem for the success of the mission.

2.3 Modules

Opt-OS has a modular structure as seen in Figure 2. Each element contained in Ops-OS contributes to the final result by providing a set of useful information to the optimiser, which can be used for calculating aspects of the satellite's design.

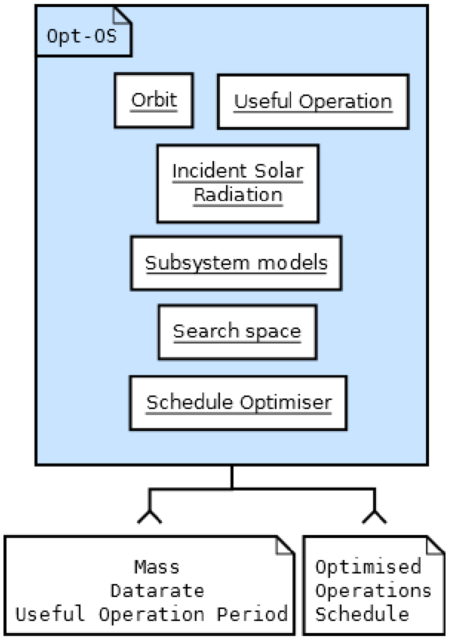


Figure 2: Opt-OS modular structure.

A description of the modules comprising Opt-OS follows.

2.3.1 Orbit and useful operation

The orbit module calculates the satellite's type of orbit, based on its classical orbital elements [14]. The resulting orbit, spanning for r revolutions, is discretised into n timesteps of equal duration. Both r and n can be chosen by the designer, based on the mission requirements. Coarser grained orbits i.e orbits comprising less thus longer timesteps comprise a smaller search space. This can lead to

faster computation yet lower schedule flexibility. Whereas finer grained orbits lead to increased computational effort yet finer schedule control. A trade-off has to be made depending on the mission objectives and computational resources available.
The useful operation module calculates the useful operation timeslot per subsystem throughout the mission timeline, based on the available subsystems onboard as well as the mission requirements. That way, an initial operations plan OP_{max} is formed corresponding to the maximum useful operation time for every subsystem on-board.

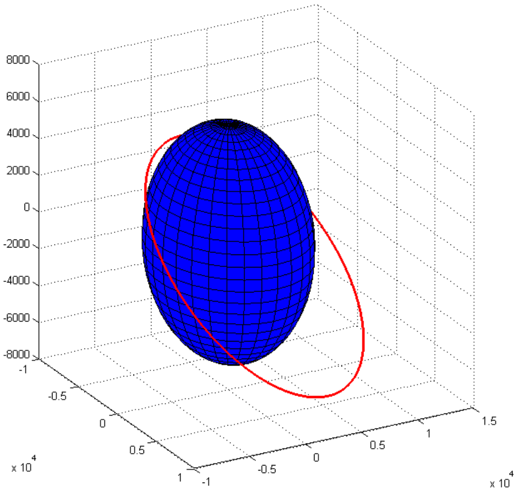


Figure 3: An example of a single orbital revolution for the chosen orbit with: a - 9105.6 km, e - 0.2444, i - 34.9398 °, $RAAN$ - 110.4872 °, ω - 90 °, M_o - 130.42 °. Based on the aforementioned orbital elements and the mission requirements, the maximum useful operation time of each subsystem per revolution is calculated.

2.3.2 Incident Solar Radiation

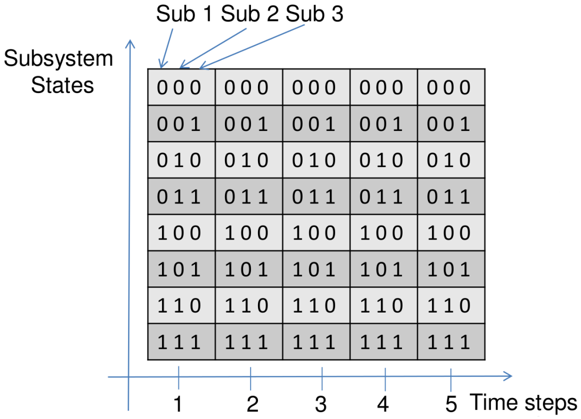
The incident solar radiation module performs analytical calculation of the intensity of direct solar radiation at any point of the satellite's surface. The solar radiation characteristics including the irradiance vector are calculated, contributing to the final calculation of the satellite solar array power output at any given time instance throughout the satellite orbit. This module can also be used for performing satellite thermal analysis, which can be used for validating the mission requirements and design or imposing design constraints.

2.3.3 Subsystem models

The subsystem models module comprises analytical models of all vital subsystems on-board the satellite, namely Attitude Control Subystem (ACS), Power subsystem, Harness, Command & Data Handling (C& DH), Thermal, Propulsion (PROP). Based on a specific set of inputs, each model can output the basic physical and electrical characteristics of each subsystem as well as design characteristics such as the solar array area, the antenna type, the downlink datarate and so on.
Model inputs are divided into two categories: *Design parameters* and *Environmental parameters*. The *design parameters* can be decided by the designer with confidence or with uncertainty, depending on the experience and previous knowledge on the respective design field. For instance the battery and solar cell type onboard the satellite can be considered part of the design parameters, as these choices depend on the designer's discretion. *Environmental parameters* cannot be decided, they are considered fixed due to environmental constraints. For example the solar flux is considered an environmental parameter since it does not depend on the designer

2.3.4 Search space

Based on the available subsystems on-board, a 2-dimensional search space is formed representing all the possible satellite operating states (Y-axis) during each timestep throughout the satellite mission time (X-axis) as seen in Figure 5.



000	000	000	000	000
001	001	001	001	001
010	010	010	010	010
011	011	011	011	011
100	100	100	100	100
101	101	101	101	101
110	110	110	110	110
111	111	111	111	111

Figure 5: Opt-OS search space. X-axis represents the mission timeline in discrete timesteps, Y-axis represents all possible subsystem operation combinations in Boolean truth values.

Each subsystem's operation state is represented by a Boolean truth value where 0 corresponds to the

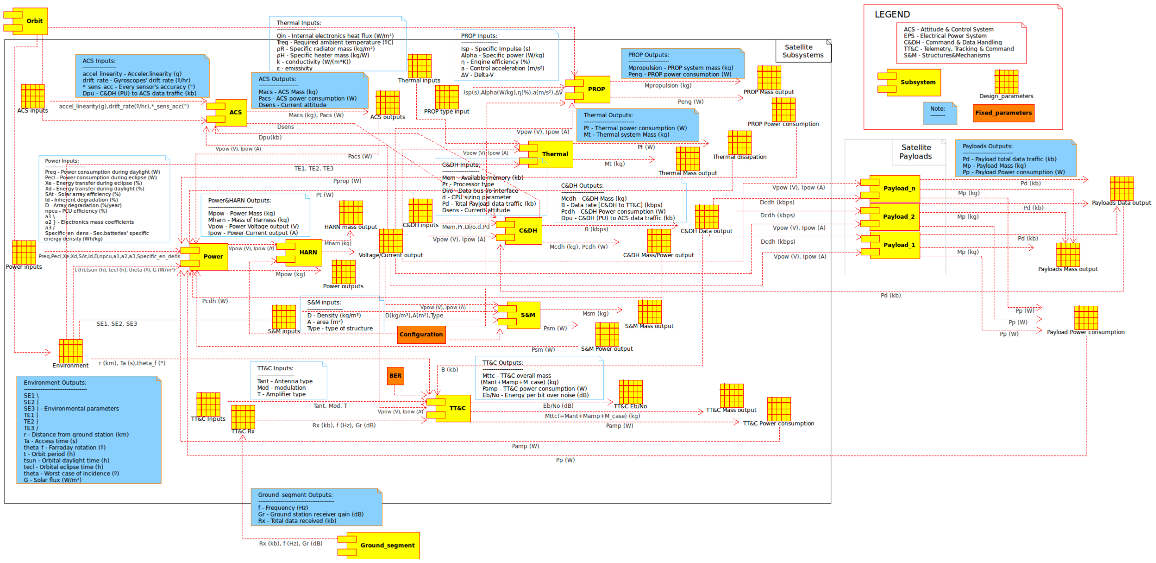

Orbit
Thermal Inputs:
Qin - Internal electronics heat flux (W/m²)
Treq - Required ambient temperature (ºC)
ρR - Specific radiator mass (kg/m²)
ρH - Specific heater mass (kg/W)
k - conductivity (W/(m*K))
ε - emissivity
PROP Inputs:
Isp - Specific Impulse (s)
Alpha - Specific power (W/kg)
η - Engine efficiency (%)
a - Control acceleration (m/s²)
ΔV - Delta-V
Satellite Subsystems
LEGEND
ACS - Attitude & Control System
EPS - Electrical Power System
C&DH - Command & Data Handling
TT&C - Telemetry, Tracking & Command
S&M - Structures&Mechanisms
Subsystem
Design_parameters
Note:
Fixed_parameters
ACS Inputs:
accel_linearity - Acceler.linearity (g)
drift_rate - Gyroscopes' drift rate (º/hr)
*_sens_acc - Every sensor's accuracy (")
Dpu - C&DH (PU) to ACS data traffic (kb)
ACS inputs
accel_linearity(g),drift_rate(º/hr),*_sens_acc(")
ACS
ACS Outputs:
Macs - ACS Mass (kg)
Pacs - ACS power consumption (W)
Dsens - Current attitude
Macs (kg), Pacs (W)
ACS outputs
Dsens
Dpu(kb)
Thermal inputs
PROP type input
Isp(s),Alpha(W/kg),η(%),a(m/s²),ΔV
PROP
PROP Outputs:
Mpropulsion - PROP system mass (kg)
Peng - PROP power consumption (W)
Mpropulsion (kg)
Peng (W)
PROP Mass output
PROP Power consumption
Thermal Outputs:
Pt - Thermal power consumption (W)
Mt - Thermal system Mass (kg)
Thermal
Pt (W)
Mt (kg)
Thermal dissipation
Thermal Mass output
Vpow (V), Ipow (A)
Pacs (W)
TE1, TE2, TE3
Pprop (W)
Pt (W)
Power Inputs:
Preq - Power consumption during daylight (W)
Pecl - Power consumption during eclipse (W)
Xe - Energy transfer during eclipse (%)
Xd - Energy transfer during daylight (%)
SAt - Solar array efficiency (%)
Id - Inherent degradation (%)
D - Array degradation (%/year)
npcu - PCU efficiency (%)
a1 \
a2 } - Electronics mass coefficients
a3 /
Specific_en_dens - Sec.batteries' specific energy density (Wh/kg)
Power inputs
Preq,Pecl,Xe,Xd,SAt,Id,D,npcu,a1,a2,a3,Specific_en_dens
t (h),tsun (h), tecl (h), theta (º), G (W/m²)
Power
Power&HARN Outputs:
Mpow - Power Mass (kg)
Mharn - Mass of Harness (kg)
Vpow - Power Voltage output (V)
Ipow - Power Current output (A)
HARN
Mharn (kg)
HARN mass output
Mpow (kg)
Power outputs
Voltage/Current output
C&DH Inputs:
Mem - Available memory (kb)
Pr - Processor type
Di/o - Data bus i/o interface
d - CPU sizing parameter
Pd - Total Payload data traffic (kb)
Dsens - Current attitude
C&DH inputs
Mem,Pr,Di/o,d,Pd
C&DH
C&DH Outputs:
Mcdh - C&DH Mass (kg)
B - Data rate [C&DH to TT&C] (kbps)
Pcdh - C&DH Power consumption (W)
Dpu - C&DH (PU) to ACS data traffic (kb)
B (kbps)
Mcdh (kg), Pcdh (W)
C&DH Data output
C&DH Mass/Power output
Satellite Payloads
Payloads Outputs:
Pd - Payload total data traffic (kb)
Mp - Payload Mass (kg)
Pp - Payload Power consumption (W)
Payload_n
Payload_2
Payload_1
Dcdh (kbps)
Pd (kb)
Mp (kg)
Payloads Data output
Payloads Mass output
Pp (W)
Payload Power consumption
Pcdh (W)
S&M inputs:
D - Density (kg/m³)
A - area (m²)
Type - type of structure
S&M inputs
D(kg/m³),A(m²),Type
Configuration
S&M
Msm (kg)
Psm (W)
S&M Mass output
S&M Power output
SE1, SE2, SE3
Environment
r (km), Ta (s),theta_f (º)
Environment Outputs:
SE1 \
SE2 |
SE3 | - Environmental parameters
TE1 |
TE2 |
TE3 /
r - Distance from ground station (km)
Ta - Access time (s)
theta_f - Farraday rotation (º)
t - Orbit period (h)
tsun - Orbital daylight time (h)
tecl - Orbital eclipse time (h)
theta - Worst case of incidence (º)
G - Solar flux (W/m²)
TT&C Inputs:
Tant - Antenna type
Mod - modulation
T - Amplifier type
BER
B (kb)
TT&C Outputs:
Mttc - TT&C overall mass (Mant+Mamp+M_case) (kg)
Pamp - TT&C power consumption (W)
Eb/No - Energy per bit over noise (dB)
TT&C Inputs
TT&C Rx
Tant, Mod, T
TT&C
Rx (kb), f (Hz), Gr (dB)
Eb/No (dB)
Mttc(=Mant+Mamp+M_case) (kg)
Pamp (W)
TT&C Eb/No
TT&C Mass output
TT&C Power consumption
Pp (W)
Ground_segment Outputs:
f - Frequency (Hz)
Gr - Ground station receiver gain (dB)
Rx - Total data received (kb)
Ground_segment

Figure 4: Full satellite blueprint based on analytical subsystem modelling. Each subsystem required a specific set of inputs, producing the basic electrical and physical subsystem characteristics amongst their outputs.

'subsystem *OFF*' state whereas 1 corresponds to the 'subsystem *ON*' state. The mission timeline is represented as a set of discrete timesteps n of equal duration. The search space is of size $(2^S \cdot n)$, with S being the number of available subsystems on-board the satellite.

2.3.5 Schedule Optimiser

The schedule optimiser is based on an Ant Colony System [6, 7, 8, 9, 10] inspired algorithm, designed to perform automated optimal operations scheduling according to the available resources on-board as well as the constraints imposed by environmental or technical parameters. A colony of digital agents performs stochastic exploration of the search space, aiming at finding the optimal route corresponding to the optimal operations schedule, setting the maximum possible operation time per subsystem as objective. The main satellite constraints are the power availability on-board as well as target visibility (e.g ground station visibility). The power availability constraint dictates the maximum amount of subsystems that can be kept ON at any time instant whereas the target visibility constraint dictates individual subsystem characteristics such as the minimum datarate allowing the satellite to downlink each full data packet during a single transmission session.

Initially the algorithm's parameters are set. The ants start exploring the search space, forming possible optimal pathways. Once all ants comprising a colony reach the end, each pathway's corresponding

Algorithm 2 Schedule optimiser algorithm

```
Set values for $m, \alpha, \rho, \xi, I_{max}$
Calculate $\tau_0$ using Eq. (6)
Define candidate list $J_i \quad \forall i \in [search\ space]$
while $I_{max}$ not met do
  Place ants on source
  for k = 1:m do
    Retrieve $\tau_{lj}$ with $l \in J_i^k$
    Decide next $j^k$ using Eq. (7)
    Perform a local pheromone update on edge $ij$ using Eq. (8)
    if all ants have reached sink then
      break
    end if
  end for
  Calculate the quality of all tours $Q^k$ using Eq. (9)
  Find $Q^+$ using Eq. (10)
  Perform a global pheromone update on $Q^+$ using Eq. (11)
  Once $I_{max}$ is met, break
end while
```

quality is measured, the best quality pathway, Q^+, is enhanced with pheromone and all ants are placed back to the starting point where they initiate a new search. This iterative process continues until the optimisation termination criterion, I_{max}, is met, forcing the optimisation cycle to end. A more detailed description of the steps included in Algorithm 2 is found below.

- Set values for the number of ants m comprising a colony, the pheromone influence parameter α, the global pheromone evaporation parameter ρ, the local pheromone evaporation parameter ξ and the maximum number of iterations allowed I_{max}.

- Distribute a uniform layer of initial pheromone, τ_0, on all edges comprising the search space. It has been found [15] that a good convention for setting the initial pheromone level τ_0 is

  $$\tau_0 = C^{-1} \tag{6}$$

  where C here is the mean of the total power consumption of all subsystems on-board the satellite throughout the mission timeline we examine.
  The initial pheromone level should allow enough iterations to take place, giving the ants enough time to converge towards an optimal solution. Care should be taken not to set the τ_0 level very high or very low though. A very high τ_0 will cancel the effect of pheromone deposition during global pheromone update at the end of each iteration, leading to an increased number of iterations before converging to a good quality solution. On the other hand, a very low τ_0 will lead the exploration process to a halt as soon as the search space pheromone level reaches zero. Since the next ant move depends on the probability distribution deriving from Eq. (7), allowing τ_{il} ($withl\ \in J_i^k$) to reach zero will result in p_{il}^k equal to zero hence ant k will stop exploring.

- Define the candidate list ,J_i, of every node in the search space with $i \in [search\ space]$.

- While the optimisation termination criterion ,I_{max}, is not met

- Place all ants on source

- For all the ants ,m, of the colony

- Decide on the next ant step ,j^k, using a variant of the probabilistic part of the *pseudo-random proportional rule*

  $$p_{ij}^k = \frac{[\tau_{ij}]^\alpha}{\sum_{l \in J_i^k} [\tau_{il}]^\alpha} \tag{7}$$

  where τ_{ij} is the pheromone level of edge ij, α is a positive weight parameter.
  The next node is selected using *stochastic sampling* also known as *roulette wheel selection method.* Utilising a weighted sample selection process, each ant chooses its next step accordingly. Higher probabilities within the probability distribution p_{il}^k acquire a bigger weight, corresponding to a larger area on the roulette wheel. This leads to a biased selection favouring the choice of nodes corresponding to higher probabilities within the distribution p_{il}^k.

- Perform a local pheromone update on edge ij after each ant move.

  $$\tau_{ij}(t) \leftarrow (1 - \xi) \cdot \tau_{ij}(t) + \xi \cdot \tau_0(t) \tag{8}$$

  where i, j are the nodes comprising edge ij that was just visited by ant k, ξ is the local pheromone evaporation rate parameter, τ_0 is the initial pheromone trail value occurring from Eq. (6)

- If all ants reached the end, break the loop in order to proceed to measuring the tours' quality.

- Calculate the quality Q^k of every path formed.

  $$Q^k = T^k \cap OP_{max} \tag{9}$$

  where T^k is the path that ant k formed throughout this iteration and OP_{max} is the maximum useful operation period for all subsystems onboard

- Find the best-so-far solution quality Q^+.

  $$Q^+ = max(Q) \tag{10}$$

- A global pheromone update is performed on Q^+

  $$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) \tag{11}$$

  where ij are all the edges comprising Q^+, ρ is the global pheromone evaporation parameter and the solution quality is calculated based on

  $$\Delta\tau ij(t) = \rho \cdot \frac{Q^+}{10^{|Magn_{t_0}|+|Magn_\rho|} \cdot P_{peak}}$$

where Q^+ is the best-so-far solution quality, $|Magn_{\tau_0}|$ is the absolute value of the magnitude of the initial pheromone level τ_0, $|Magn_\rho|$ is the order of magnitude of the global pheromone evaporation parameter ρ and P_{peak} is the peak power consumption found in OP_{max}.

- Once the optimisation criterion ,I_{max}, is met, end the optimisation cycle.

Two important factors have significant influence over the quality of the optimiser's solution as well as the computational effort, namely the number of ants ,m, per colony and the global pheromone evaporation parameter ,ρ. As seen in Figure 6, a higher m can improve the solution quality for the same number of iterations. As expected in serial agent-based processes, the computational time increases with the number of ants.

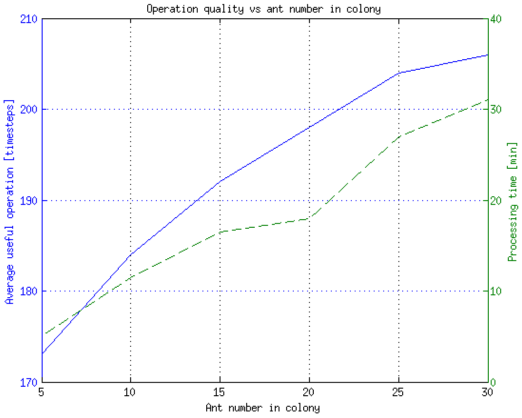


Figure 6: Opt-OS solution quality versus processing time as a function of ants per colony. The I_{max} was set to 1000 iterations. Maximum feasible solution quality is 206.

The global pheromone evaporation parameter ρ influence as seen in Figure 7 is noteworthy. A lower ρ allows the ants to explore the search space more freely avoiding being trapped in local optima too easily. The enhanced exploration leads to a higher computational time as more options need to be explored, yet a significantly improved solution quality.

3 TEST CASE

In this section, we demonstrate the flexibility of Opt-OS by utilising it in the following proposed satellite design integrated approach described in Figure 8. A simple generic satellite is considered, containing an Electrical Power Subsystem (EPS) and a Telecommunications subsystem as described in [16].

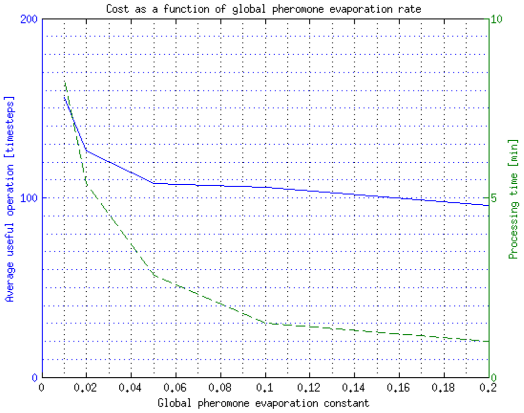


Figure 7: Opt-OS solution quality versus processing time as a function of global pheromone evaporation. The I_{max} was set to 1000 iterations. The m was set to 10 ants.

The design process is performed by two enclosed loops, namely the Inner loop (Opt-OS) and the Outer loop (MACS). The Inner loop models the satellite and derives its optimal operations schedule as described above. The Inner loop's output comprises three main quantities, the total satellite mass ,M_{tot}, the satellite downlink datarate ,D_{rate} and the optimal mission operations schedule quality ,Q^+, i.e the cardinality of the intersection between the maximum useful operation period ,OP_{max}, and the current optimised operations schedule based on the quality criterion of Eq. 9. Depending on the mission requirements, the Inner loop can output more results like for example the onboard databus datarate and so on.

The Outer loop comprises MACS, a stochastic multi-objective optimization algorithm combining together a number of heuristics as described in [5]. Using a combination of heuristics, a population of agents explores the search space both in a global fashion and around the neighbourhood of each agent. The heuristic driven optimisation process is complemented with a local and global archive. The algorithm comprises eight basic steps namely Initialization, Collaboration, Selection, Filtering, Repulsion, Local actions, Hyperrectangle update, Archive update and the Stopping rule.

The Outer loop utilises the Inner loop's outputs, passing them to MACS. Furthermore, the upper and lower bound of all subsystem design parameters is set. A multi-objective optimization process takes

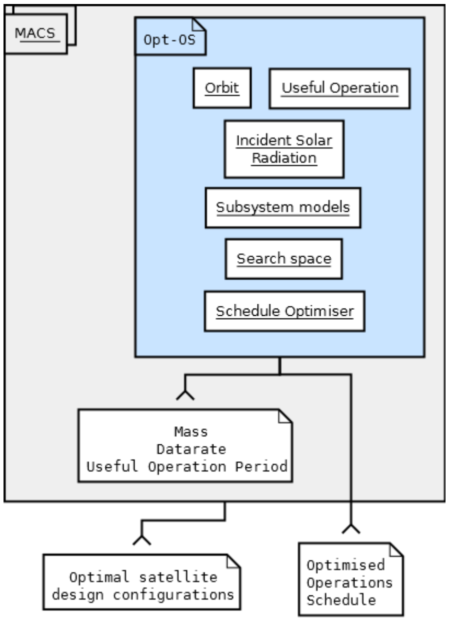


Figure 8: Integrated system-operations design approach.

place where the satellite design parameters are optimised depending on the objectives set. In this test case the objectives set are the following: $min(M_{tot})$ while $max(D_{rate})$ and $max(Q^+)$.
The Outer loop finally outputs a Pareto set of points as seen in Figure 9, representing optimisers. Each point represents a vector of optimal subsystem design parameters leading to a full optimal satellite design.

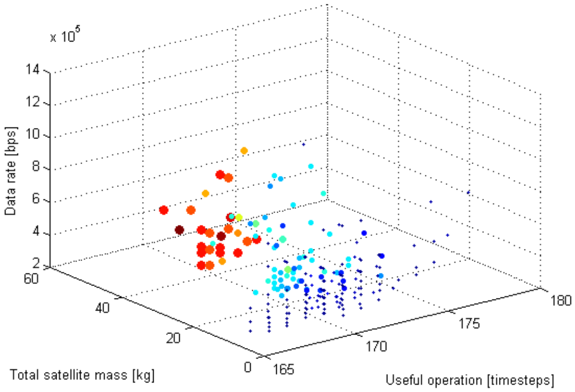


Figure 9: Pareto set of optimisers. Each point corresponds to a full optimal satellite design. Thicker points correspond to higher satellite masses.

The table below contains an example of 5 optimal satellite design parameter sets corresponding to Pareto points lying in the lower mass spectrum.

Table 1: An example of five optimal design parameter sets lying in the lower mass spectrum of the Pareto set.

	1	2	3	4	5
TTC design parameters					
f [GHz]	4	3.87	4.21	4	5.15
Ant	2	2	2	2	2
Mod	0.02	0.46	0.3	0.02	0.28
T	0.44	0.53	0.56	0.44	0.69
Te [K]	91.49	91	86.91	91.49	93.77
F [dB]	12.37	11.6	13.6	12.37	10.24
Tet [K]	49.62	53.57	49.62	66.45	77.58
Ft [dB]	12.79	13.15	12.21	12.79	13.78
$Tant$ [K]	50.98	66.1	62.66	62.45	69.13
n_t	0.62	0.77	0.63	0.75	0.55
Lc [m]	0.3	0.3	0.3	0.3	0.3
EPS design parameters					
SA_t	0.27	0.29	0.25	0.27	0.32
Xe	0.63	0.64	0.62	0.63	0.6
Xd	0.8	0.85	0.82	0.8	0.81
Id	0.85	0.86	0.7	0.85	0.83
S_M $[kg/m^2]$	2.64	1.36	2.13	2.64	2.72
n_{pcu}	0.89	0.9	0.86	0.89	0.9
SED $[Wh/kg]$	123.3	77.15	63.81	123.3	183.5
$Vdrop$	0.04	0.03	0.04	0.04	0.04

The TTC design parameters correspond to: **f** - Transmission Frequency $[GHz]$, **Ant** - Antenna Type [1 - Horn, 2 - Parabolic reflector], **Mod** - Modulation (Treating the modulations range as continuous gives the designer the ability to choose any type of existing or possible future modulations in the form of real numbers interpolated over the modulation range [14]), **T** - Amplifier Type (Treating the amplifier types range as continuous gives the designer the ability to choose any type of existing or possible future amplifier types in the form of real numbers interpolated over the amplifier types range [14]), **Te** - Amplifier noise $[K]$, **F** - Receiver Noise Figure $[dB]$, **Tet** - Amplifier noise $[K]$, **Ft** - Transmitter Noise Figure $[dB]$, **Tant** - Antenna Noise Temperature $[K]$, **nt** - Antenna efficiency [%], **Lc** - Antenna characteristic length $[m]$.
The EPS design parameters correspond to: **SAt** - solar cell efficiency [%], **Xe** - Energy transfer during eclipse [%], **Xd** - Energy transfer during daylight [%], **Id** - Inherent degradation [%], **SM** - array specific mass $[kg/m^2]$, **npcu** - PCU efficiency [%], **SED** - Secondary batteries specific energy density $[Wh/kg]$,

Vdrop - Maximum permissible bus voltage drop [%].

4 CONCLUSION

In this paper, a flexible optimal operations scheduling approach is proposed. A modular tool is presented, offering automation of the operations scheduling process while enhancing the robustness of the operations schedule. The tool, namely Opt-OS, comprises a set of modules modelling the satellite and its environment as well as a nature-inspired optimiser outputting the optimal operations schedule on the spot. Opt-OS can be used in order to alleviate the workload of mission operations teams, enhance the robustness of the operations schedule, devise emergency operations plans. All in all, Opt-OS aims at maximising the use of the satellite throughout each mission timeline instant.

Furthermore, Opt-OS can be used as the basis of an integrated automated optimal satellite design approach, where initially an optimal operations schedule is designed followed by a set of optimal satellite designs utilising this operations schedule. This automated approach can offer enhanced robustness over the classical design approach while outputting a cost effective design produced in a timely fashion, thus making space accessible to a wider spectrum of institutions with lower budgets.

5 ACKNOWLEDGEMENTS

The author would like to thank Mr. Alasdhair Beaton and Mr. Massimo Vetrisano for their valuable contribution to this work. This research has been partially conducted using the University of Strathclyde High Performance Computer (HPC).

REFERENCES

[1] Harland, D. M. and Lorenz, R. D., *Space Systems Failures, Disasters and Rescues of Satellites, Rockets and Space Probes*, Springer and Praxis, 2005.

[2] "Mars Climate Orbiter Mishap Investigation Board Phase I Report," Tech. rep., 1999.

[3] "Lewis Spacecraft Mission Failure Investigation Board: Final Report," Tech. rep., 1998.

[4] Sims, M. R., *Beagle-2, Lessons Learned and Management and Programmatics*, University of Leicester, 2004.

[5] Vasile, M. and Locatelli, M., "A hybrid multiagent approach for global trajectory optimization," *Journal of Global Optimization*, Vol. 44, No. 4, 2009, pp. 461–479.

[6] Dorigo, M. and Stuetzle, T., *Ant Colony Optimization*, MIT Press, 2004.

[7] Dorigo, M. and Gambardella, L., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transaction on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53–66.

[8] Dorigo, M. and Gambardella, L., "Ant Algorithms for Discrete Optimisation," *Artificial Life, MIT Press*, Vol. 5, No. 2, 1999.

[9] Bonabeau, E., Dorigo, M., and Theraulaz, G., *Swarm Intelligence From Natural to Artificial Systems*, Santa Fe Institute: Studies in the Sciences of Complexity, Oxford University Press, 1999.

[10] Ostfeld, A., editor, *Ant Colony Optimization - Methods and Applications*, InTech, 2011.

[11] Papadimitriou, C. H., "The Euclidean travelling salesman problem is NP-complete," *Theoretical Computer Science*, Vol. 4, No. 3, 1977, pp. 237 – 244.

[12] Karlson, P. and Luscher, M., "Pheromones: a New Term for a Class of Biologically Active Substances," *Nature*, , No. 183, 1959, pp. 55–56.

[13] Cover, T. M. and Hart, P. E., "Nearest Neighbor pattern classification," *IEEE Transactions on Information theory*, Vol. IT-I3, No. 1, 1967, pp. 21–27.

[14] Larson, W. J. and Wertz, J. R., *Space Mission Analysis and Design*, Microcosm Press and Kluwer Academic Publishers, 2005.

[15] Komninou, E., Vasile, M., and Minisci, E., "Optimal power harness routing for small-scale satellites," October 2011.

[16] Vasile, M., Minisci, E., Zuiani, F., Komninou, E., and Wijnands, Q., "Fast evidence-based space system engineering," .