17rd International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 354

Autonomous Space Operations Planner and Scheduler (ASOPS): Optimal and Autonomous Operations in Space

Francesco Porcelli *, Johannes Bachmann, Mohd Bilal, Artur Kinzel, Emir Gadzo, Maren Hülsmann, Tom Andert

Institute of Space Technology and Space Applications, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
*Corresponding Author (francesco.porcelli@unibw.de)

Abstract

Planning and execution of space operations can be quite challenging when it comes to dealing with multiple payloads, all having different goals, requirements, and constraints. Not all payloads can operate at the same time due to limited resources on-board a spacecraft, such as available power, data storage and downlink capacity. The complexity of future space missions will likely increase in terms of operations to be performed and reliability aspects with respect to mission success. Moreover, there is potentially a need for more autonomous, flexible, and optimized operations conducted on-board the spacecraft to increase the efficiency of resources used and the overall scientific outcome of future space missions.

The Autonomous Space Operations Planner and Scheduler (ASOPS) is planned as an on-board system that aims to provide optimized schedules and autonomous space operations by taking advantage of the increased data availability and computational capacity on-board future spacecraft. ASOPS and its features are currently in development for the Athene-1 satellite within the framework of the Seamless Radio Access Network for Internet of Space (SeRANIS) Project, where a multitude of innovative experiments will be performed in space. The SeRANIS Project provides with Athene-1 a perfect environment to deploy, test, and demonstrate ASOPS and its capabilities. Within the ASOPS project, state-of-the-art spacecraft simulation, data-based system modeling, optimization, and AI-based techniques will be studied to evaluate the performance and benefits of autonomous planning and scheduling.

In this paper we present the concept and architecture of ASOPS and introduce a promising method for optimization of schedules on the use-case of Athene-1. Additionally, we show first results in the development of the schedule optimization.

Keywords: Space Operations, Autonomy, Optimization, Artificial Intelligence

Abbreviations

AI-OBC	Artificial Intelligence capable On-Board Computer
ASOPS	Autonomous Space Operations Planner and Scheduler
A3C	Asynchronous Advantage Actor
CDL	Constraint Definition Language
DQN	Deep Q-Networks
ECSS	European Cooperation for Space Standardization
EOS	Earth Observation Satellites
EO-1	Earth Observing One
GDL	Goal Description Language
GS	Ground Station
LEO	Low Earth Orbit
LSA	Local Search Algorithm
MDL	Mission Description Language
MPF	Mission Planning Facility
RL	Reinforcement Learning
SC	Spacecraft
SDL	Spacecraft Description Language
SeRANIS	Seamless Radio Access Network for Internet of Space
TC	Telecommand
TM	Telemetry
UniBw M	Universität der Bundeswehr München

1 Introduction

Spacecraft operations and mission planning have gradually evolved and built up on experience of operation centers from several missions throughout the years. This experience has ensured the availability of complex and robust planning software tools. These software generally receive flight dynamics inputs as two-line-elements and orbit/event files, ground station availability information, and user requests for payload. Including detailed models of the environment, spacecraft, and payload, these tools process the data and generate a schedule in the form of a conflict free commands file that is subsequently uplinked to the spacecraft. Although very reliable and robust, these planning and scheduling concepts need a connection with the spacecraft to uplink the commands of the schedule computed. Additionally, ground-based planning tools rely on predictions of the spacecraft state based on simulations without having real-time access to the current state. The scheduling of spacecraft operations is challenging when dealing with multiple payloads, all having different goals, requirements, and constraints while on-board resources are limited. In addition to payload constraints, mission and platform constraints have to be considered in the planning of operations.
With the Autonomous Space Operations Planner and Scheduler (ASOPS) we explore and demonstrate innovative solutions for the planning and scheduling problem.
This paper is organized as follows: Chapter 2 describes the motivation, the concept, architecture, and the design of ASOPS; in Chapter 3 first results of optimized schedules are shown and the next steps that are foreseen in the development are outlined; conclusions are given in Chapter 4.

2 ASOPS

The Autonomous Space Operations Planner and Scheduler (ASOPS) is an on-board system that creates optimized schedules and enhances autonomous space operations. ASOPS is developed with the goal of increasing the level of on-board autonomy of the spacecraft and optimizing the resource utilization, which is a limiting factor in small satellite missions. Although space missions such as Deep Space 1 and Earth Observing One (EO-1) have already demonstrated AI-based autonomy architectures in space, they could only execute a portion of the autonomous capabilities intended due to reliability and robustness aspects, i.e. trust [1]. ASOPS applies state of the art methods and concepts to advance the optimization of schedules and the level of autonomy for spacecraft operations one step further in order to support upcoming deep space and multi-constellation missions.
As an on-board planning tool, ASOPS is able to directly command the spacecraft. This improves the responsiveness to unforeseen events by autonomous adjustment of the on-board schedule. ASOPS aims to optimize the schedule by using innovative methods supported by a deep knowledge of the platform and payload behavior. Additionally, the direct access to the current state of the spacecraft resources allows a further improvement in the optimization by a continuous adaptation of the prediction models. Different simulations and optimization methods are used to deliver a schedule for spacecraft operations with optimal resource utilization and maximized scientific outcome considering all goals, requirements, and constraints.

ASOPS is developed by researchers at the Universität der Bundeswehr München (UniBw M). Its on-board software will be part of the small satellite Athene-1, developed in the framework of the Seamless Radio Access Network for Internet of Space (SeRANIS) project [2]. SeRANIS focuses on the analysis of the feasibility and maturity of key technologies in space and offers ASOPS the opportunity to demonstrate its capabilities. ASOPS is part of a multitude of experimental payloads on-board Athene-1, which is scheduled to launch in 2025 into a frozen Sun Synchronous Low Earth Orbit (LEO). SeRANIS shall also act as an innovation hub and provide open interfaces for start-ups and industry to test new technologies for space applications, where ASOPS is a part of [2, 3].

The following Chapters provide the overview of the process behind ASOPS, outline the autonomy aspects considered, and the architecture of ASOPS defined.

2.1 Process Flow Overview

As illustrated in Figure 1, on a high-level perspective the process flow of ASOPS starts with the acquisition of data, where Telecommand (TC) and Telemetry (TM) data are collected and pre-processed on-board the spacecraft. Following the data collection and pre-processing, ASOPS determines an optimized schedule for dedicated spacecraft operations. After an optimized schedule is determined, ASOPS evaluates the schedule via an AI-based autonomy supervisor that is responsible to assess the schedule with respect to reliability, robustness, and safety aspects. The last process is dedicated to post-process the schedule generated by ASOPS. The post-processing includes the conversion of the schedule into readable TC that can be executed by the spacecraft. Further details about the autonomy aspects for ASOPS considered and the architecture of ASOPS defined are outlined in Chapter 2.2 and 2.3 accordingly.

2.2 Autonomy

The subject of reliable and robust autonomy in space applications is vital since the beginning of the space exploration and shows promising development due to missions such as Deep Space 1 and EO-1 in recent years [1]. Nevertheless, further development of

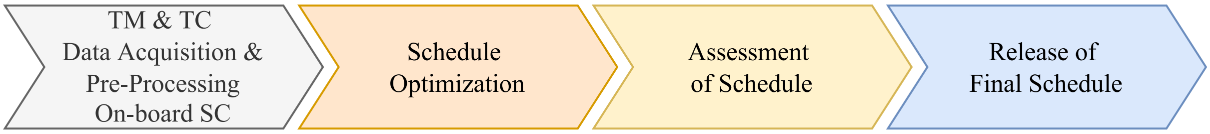


Figure 1: ASOPS High-Level Process Flow

full-autonomy for spacecraft operations is still a current topic.

According to Jonsson et al. [4] autonomy and automated processes are defined as the ability of the system to make rational, informed, self-determined, and self-reliant decisions. This includes the ability to sense the environment and the awareness of its own capabilities and actions. In addition, such a system should respond to non-nominal situations by autonomously adapting its sequence of action.

Another definition is given by Truszkowski et al. [5] where they are stating that processes are autonomous or automated if executed without human intervention. In case of spacecraft operations, autonomous processes involve decision-making and emulation of human behaviour. For example, within spacecraft autonomy and scheduling this involves autonomously identifying slots for ground station communication, establishing links, or autonomously performing uplink and downlink activities [5].

Furthermore, according to the European Cooperation for Space Standardization (ECSS) (ECSS-E-ST-70-11C [6]), which outlines guidelines and requirements for the design of on-board functions for autonomous spacecraft, autonomy levels can be separated into three main sections. These sections are differentiated into Nominal Mission Operations, Mission Data Management, and On-Board Fault Management. The guidelines given in [6] shall ensure that on-board functions can be executed in any nominal or predefined non-nominal mode without relying on ground segment intervention. The section including guidelines and levels of autonomy that are most relevant for ASOPS are outlined within the Nominal Mission Operations, as listed in Table 1.

Table 1: Guidelines for Autonomy for Nominal Mission Operations according to [6]

Autonomy Section	**Autonomy Levels**
Nominal Mission Operations	1. execution mainly under real-time ground control 2. execution of pre-planned mission operations on-board 3. execution of adaptive mission operations on-board 4. execution of goal-oriented missions operations on-board

The definitions of autonomy by Jonsson [4] and Truszkowski [5] together with the guidelines outlined by the ECSS standard on space segment operability [6] give a baseline of the autonomous capabilities of ASOPS. These are formulated as follows:

- awareness of the environment and spacecraft general capabilities
- monitoring of all spacecraft subsystems states, and onboard activities
- decision-making if the schedule generated is reliable and robust enough to be executed
- adaption of the schedule in case of non-nominal situations

2.3 Architecture

The ASOPS software can be broadly grouped into two components, namely, the ASOPS-Core and the ASOPS-Interfaces, as shown in Figure 2. To ensure flexibility and adaptability for the space market, ASOPS is designed in a modular way, such that the software can be augmented to run on different satellite platforms and for different missions. ASOPS-Core provides the foundational framework to develop user defined planning and scheduling agents. The ASOPS-Interfaces provide the means to define mission specific needs and constraints. Additionally, the ASOPS-Interfaces handle interactions with the entities external to the ASOPS agent, for instance spacecraft on-board computer and space environment.

2.3.1 ASOPS-Core

This Chapter describes the various logical elements within ASOPS-Core.

Scheduler The Scheduler is the module that implements algorithms to select a set of actions based on the estimated state of the spacecraft and the environment. The Scheduler is at the center of ASOPS-Core where all the information about the mission, payload, spacecraft, and environment flows in a format suitable for the scheduler to handle. The scheduler can be single step or multi-step. A single step scheduler only selects the action or set of actions to be taken at the next time step while a multi-step scheduler selects the schedule over a longer horizon. The output of the scheduler is a set of timestamped optimal actions to be performed over the scheduling horizon.

State Estimator The State Estimator provides the scheduler with an estimate of the current state of the spacecraft and the environment. An estimate of the state is a fusion of sensor readings (obtained through spacecraft telemetry) and predicted state. The state estimator is akin to a Kalman Filter.

State Predictor The State Predictor is a state integrator that takes as input the current state and an action, from which it predicts the next state of the spacecraft and the environment. The predictor depends on environment models and a spacecraft model. The discrete state predictor is described by Eq. 1 where x_k is the state at kth step and a_k is the action or set of actions performed at step k. The function f is informed by the spacecraft and environment models. The state predictor is in a closed loop with the scheduler.

$$x_{k+1} = f(x_k, a_k) \tag{1}$$

Adaptive Spacecraft Model The Adaptive Spacecraft Model is a time dependent behavioural model of the spacecraft, e.g., the charging and discharging model of the battery, mode dependent power consumption of equipment and instruments, and data rates. The model is adaptive in the sense that the parameters get updated based on spacecraft telemetry. A familiar example is the charging capacity of the battery which degrades over time. In such a case, the model parameters need to be updated to get the correct state prediction.

Environment Models The Environment Models provide the models of the space environment. They include, among others, the gravitational and the the atmospheric model of the Earth, the solar radiation model, and geomagnetic field model. In addition to providing the forces needed to predict the system state, environment models also emit predictable events such as start and end of ground contact or eclipse times. The events are handled by the state predictor and also pipelined to the scheduling agent to take an appropriate action based on the event.

Goal Description Interpreter The Goal Description Interpreter is the interpreter for the goal description language used by ASOPS. It interprets the scientific goals of the mission and payloads and provides it to the goal handler which transforms the goals to a mathematical or structural form suitable for the scheduler. These goals drive the scheduler to produce a schedule that minimizes the distances to the goals. E.g., a mission goal could be to operate all payloads in a fair manner while a payload goal would be to maximize the number of pictures taken by the camera.

Constraint Description Interpreter The Constraint Description Interpreter is the interpreter for the constraint description language used by ASOPS and works in tandem with the constraint handler in a manner similar to the goal description interpreter. Constraints are divided in three different types of constraints: payload, platform, and mission constraints. Examples of payload constraints are pointing requirements, experiment runtime, experiment start time. Platform constraints are for example available resources on-board and attitude constraints. Examples of mission constraints include mission lifetime, predefined schedules, mission ground station constraints.

Mission Description Interpreter The Mission Description Interpreter is the interpreter for the mission description language used by ASOPS and works in tandem with the mission definition handler in a manner similar to the goal description interpreter. It must be noted that mission constraints can also be handled by the mission description interpreter and passed on to the constraint handler.

Autonomy Supervisor The Autonomy Supervisor is the agent that performs a risk assessment aimed to identify any criticality in the schedule generated with respect to safety aspects. Once the scheduler determines a optimal solution for scheduling spacecraft operations, the schedule is given to a supervising agent that performs the task of evaluating the schedule and passes it to the ASOPS Output interfaces for the creation of the corresponding TCs. This supervising agent, using an AI-based approach, shall be able to autonomously evaluate the schedule with respect to factors such as efficiency, generation of results, consideration of enquires, and reliability. In case the supervisor reject the schedule received, a feedback loop to the Scheduler is used for improving the solution of the new schedule generated.

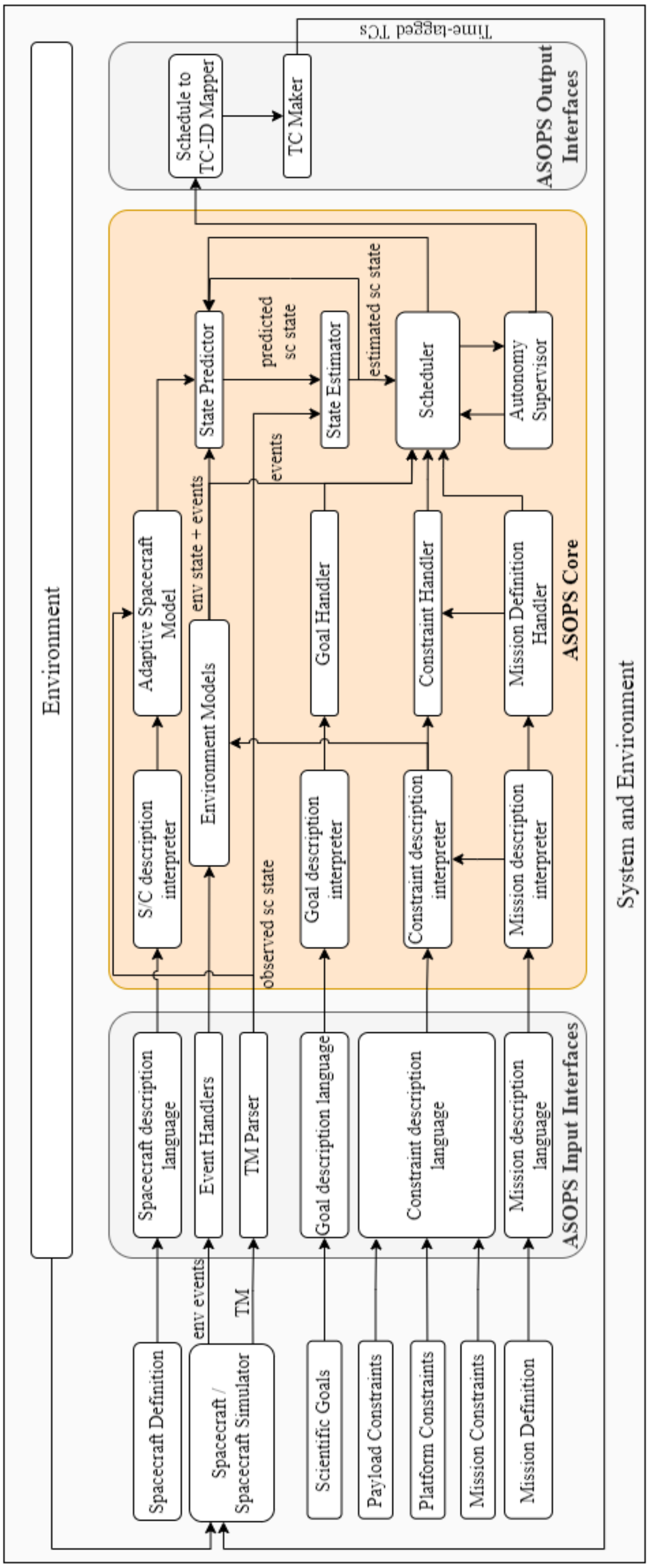


Figure 2: ASOPS Architecture

2.3.2 ASOPS Interfaces

This Chapter describes the various logical elements within ASOPS-Interfaces.

ASOPS-Interfaces assist the user in providing information required by the Scheduler, and provide outputs that can be easily transformed into user defined telecommand structures. In order to produce an optimized schedule, different information have to be taken into account:

- limitations of the spacecraft platform itself
- constraints introduced by the payload e.g. pointing, times and locations where no observation is possible
- higher level scientific objectives to be accomplished during the mission life time
- mission related restrictions
- predefined schedules that are accounted for in mission constraints

Therefore, the following ASOPS interfaces are hereby defined to support the definition of aforementioned information.

Spacecraft Description Language (SDL) The SDL is an ASOPS domain specific language that supports the definition of a spacecraft in a tree-like structure. Properties like modes, power consumption, and data rates can be defined using SDL.

Goal Description Language (GDL) The GDL supports the definition of scientific goals for payloads and the overall mission itself. A goal is a quantifiable value as described above and can be attributed to a component or an action.

Constraint Definition Language (CDL) The CDL is also an ASOPS domain specific language through which ASOPS users can describe the constraints imposed by payloads, platform, and the mission on the final schedule.

Mission Description Language (MDL) The MDL supports the definition of the mission itself. A mission definition provides information like initial orbital parameters, ground station information, and number of spacecraft.

The complete syntax and description of the description languages used by ASOPS is beyond the scope of this work and will be presented in the future.

TM Parser The TM Parser is a set of functions implemented by the users of ASOPS that can parse the telemetry of the spacecraft and create ASOPS Spacecraft State objects. These functions are clearly spacecraft dependent and therefore ASOPS does not implement any TM Parsers and only provides the interfaces to create relevant objects and load telemetry to ASOPS-Core database.

Schedule to TC-ID Mapper The Schedule to TC-ID Mapper maps the internal representation of the schedule generated by ASOPS to the TC IDs of the specific spacecraft. The final TC are generated by the TC Maker which loads Telecommand bytes from spacecraft TC database.

2.3.3 Data Flow Architecture

As outlined in Chapter 2.3.2, ASOPS has different interfaces that allow for the data acquisition required to generate an optimized schedule and provide relevant feedback to the user. The main interface elements to the ASOPS system include the Ground Station (GS) and the Spacecraft (SC) platform, as illustrated in Figure 3. On-board the spacecraft platform, ASOPS uses the current TM of payloads together with the schedules and parameters uplinked from ground to generate optimized schedule for spacecraft operations. The Figure 3 refers to the use case provided by the SeRANIS project, where ASOPS is foreseen to run on a dedicated Artificial Intelligence capable On-Board Computer (AI-OBC), which is required in order to operate the methods for optimization and autonomy of ASOPS. As an example, Figure 3 outlines different payloads that are operated on-board Athene-1.

Since ASOPS is foreseen to also manage predefined schedules, the schedules created on ground for spacecraft operations are part of the inputs as well. Furthermore, payload, platform, and mission constraints can be updated from ground. The schedules and the constraints are uplinked from the GS to the SC platform in form of TC. Once the schedules and parameters are received by the SC platform, they are forwarded to the Payload Control Subsystem that is responsible for the execution of payload operations.

With reference to Figure 2, a typical ASOPS agent deals with several different types of data ranging from Spacecraft Telemetry to timestamp data and schedules uplinked from ground in form of TC. ASOPS also performs several operations on this data leading to a complex data flow. The User Application is the starting point which implements ASOPS-Interfaces and provides goals, constraints, mission definition, environment information, spacecraft definition, and spacecraft TM to the rest of the processes. Constraints and goals are used to build constraints and goals objects. Spacecraft definition meanwhile is used to build spacecraft object model which is used to propagate the spacecraft state. Environment information is extracted from mission definition and is

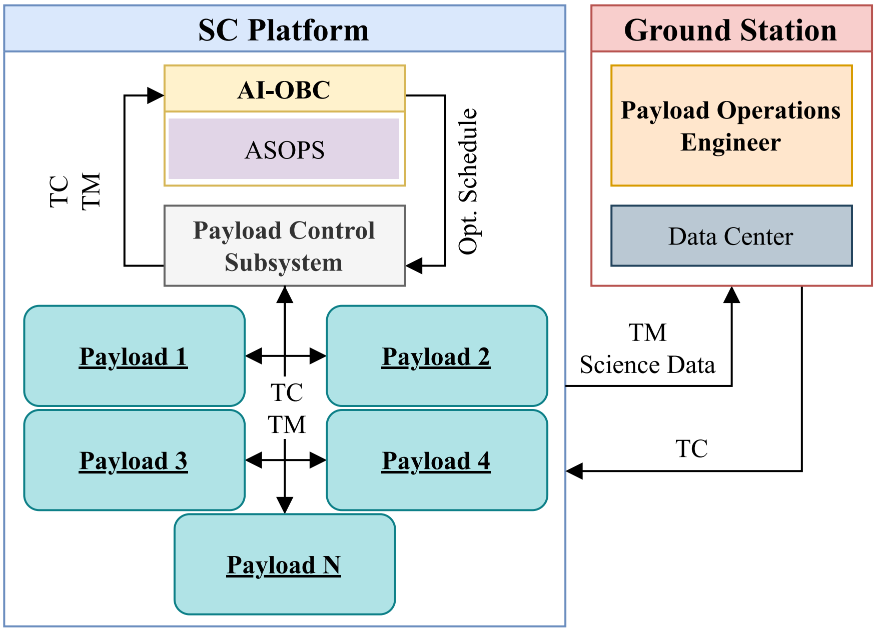


Figure 3: ASOPS Data Flow Diagram as foreseen for the SeRANIS Project

used to propagate the environment state. In the same step, predicted environment and spacecraft state are combined into a single state vector and sent forward for state estimation where spacecraft TM is used as observations to correct predictions. Finally the select actions processes receives constraint objects, goal objects, state estimate and any events emitted by the environment to select one or more actions. This process then happens in a feedback-loop with the propagate state function. If the horizon of the schedule is finished, the schedule is compiled and sent for a criticality and safety assessment and following clearance is saved to ASOPS Schedule Database. The User Application is subscribed to schedule update events from ASOPS Schedule Database and upon reception of these events, it forwards the time-tagged telecommands to the spacecraft on-board computer. The User Application may optionally also save application level TM.

2.4 *Scheduler Algorithms*

Satellite scheduling problems are characterized by a high level of complexity due to the elevate number of variables and the constraints present in the problem. Their complexity hints toward the application of heuristic and meta-heuristic approaches in order to efficiently find good solutions for practical applications [7]. The large solution space of such problems is especially relevant when the computational resources are limited, as it is the case for computers on-board satellites.

Chien et al. [8] demonstrate a timeline-based scheduling of space operations on the EO-1 spacecraft based on a Greedy Algorithm. This heuristic approach does not result in an optimal solution but it is not computationally intensive.
Lemaitre et al. [9] implement and compare different methods used to solve the selecting and scheduling observation problem for Earth Observation Satellites (EOS). The Greedy Algorithm they implement is a sequential greedy algorithm, characterized by a simple formulation and high speed. It is though not able to consider stereoscopic requests (involving coupling constraints between non consecutive images). Lemaitre et al. [9] develop an Local Search Algorithm (LSA) for selecting and scheduling observation for satellites. In this study [9], the LSA algorithm is the one that has overall better performances. Globus et al. [10] compare the performances of Hill Climbing, Simulated Annealing, and two different Genetic Algorithms in EOS scheduling problem. Sarkheyli et al. [11] model the scheduling problem for LEO satellite using the Graph Coloring Theory and then solve the problem applying a Tabu Search Algorithm. The research show the efficiency and the reduced computational time of the developed algorithm. It is though not proven with real world problems. Xhafa et al. [7] present a survey of optimization problems and resolution methods in satellite scheduling and spacecraft operations, where they evaluate several methods as Hill Climbing, Simulated Annealing, Tabu Search, and Genetic Algorithms. Genetic Algorithms are commonly used for problem optimization in the field of satellite operations [12, 13, 14, 15, 16, 17, 7, 18]. In analogy with natural selection, an initial population is generated and individuals associated to better states are mated in order to generate a more "fit" population. The fitness of individual is evaluated through a "fitness function" that is problem-dependent. Weber et al. [12] show that one year mission can be saved for a low orbit Moon mapping mission by using an evolutionary optimization tool which combines the Genetic Algorithm with two more evolutionary algorithms: Evolutionary Strategies and Differential Evolution. Sun et al. [14] construct a Mixed Integer Programming Model to formulate the scheduling problem of satellite imaging missions. The problem is encoded so that each chromosome represents a solution. A

roulette wheel selection is applied to the population whose fitness is evaluated as a weighted sum of different mission objectives. Baek et al. [15] develop a Genetic Algorithm for optimization of autonomous satellite operations. The fitness function used in the Genetic Algorithm takes into account 8 "scheduling parameters" for the problem studied: priority, deadline, profit, area, emergency, energy consumption, memory consumption, and weather conditions. Xhafa et al. [16] implement a Genetic Algorithm for a single ground station scheduling problem while Russel et al. [13] show the effectiveness of Genetic Algorithms for ground station scheduling in general. Ntagiou et al. [19] propose an automated Mission Planning System based on the Ants' Foraging Mechanism.

Previous research in the field indicates Genetic Algorithms as good candidate for finding a solution to the scheduling problem. For ASOPS, a Genetic Algorithm has been implemented and is described in Chapter 3.

Another promising approach to solve the scheduling problem is based on Reinforcement Learning (RL) methods. In RL, an agent learns how to behave through trial-and-error interactions with a dynamic environment. The actions the agent takes are decided by a policy. The goal of RL algorithms is to find an optimal policy, i.e., a policy that maximizes the reward of the agent over the long-term. Recently, deep neural networks have proven to be efficient for finding policies. Several deep-RL algorithms are actively studied to solve complex sequential decision-making problems. Among the best-known methods are algorithms such as Deep Q-Networks (DQN), Rainbow [20], policy-based algorithms such as REINFORCE [21] or actor-critic methods such as Asynchronous Advantage Actor (A3C) [22] or Proximal Policy Optimization (PPO) [23]. Hadj-Salah et al. [24] propose an acquisition scheduling approach for earth observation constellations based on Deep Reinforcement Learning. They approach constellation scheduling by having an orchestrator responsible for request ranking towards each Mission Planning Facility (MPF). The orchestrator analyzes a large-scope of data (e.g., weather forecasts, access opportunities) to optimize the global schedule, while each MPF has a narrowed and short term vision of their single satellite scheduling. They concentrate their experiments on the A3C [22] algorithm which has shown to give the best results. In a series of simulation-based experiments, the proposed method challenges the state-of-the-art heuristics.

Considering the promising potential of RL, a scheduler based on it is also currently under investigation.

3 First Results and Future Development

As mentioned in Chapter 2.4, Genetic Algorithms have already been used for optimization of spacecraft operations. For this reason, a Genetic Algorithm has been implemented for ASOPS in order to evaluate its performances and compare it with other optimization methods under investigation within this project.
The Genetic planner developed for ASOPS follows the standard paradigm of Evolutionary Algorithms. Its high level behaviour can be summarized in the following points:

- an initial population is instantiated with randomly created (and hence flawed) individuals
- the population is evolved through the application of genetic operators on selected individuals (parents), generating the children, and creating a new population
- populations are evolved until a predefined condition is met

A brief description of the model used in ASOPS is here presented, followed by its results.

3.1 Model

Every Genetic Algorithm is defined by the model used, characterized itself by: gene, chromosome, fitness function, genetic operators, and the algorithm itself.

Gene The gene represents the way the problem is encoded. In the Genetic Algorithm developed for ASOPS, the gene is an atomic structure containing 2 parameters: the *ID* of the action (or operation), and the *start_time* of such operation. At the beginning of the algorithm, an action *ID* is randomly selected from the database of defined actions and it is paired with a *start_time* randomly generated from the interval $[0, final\_time]$; where *final_time* is the end of the scheduled operations.

Chromosome The chromosome is a list of genes ordered by their *start_time* value. The number of actions to be performed in the optimal operational plan is clearly not known a priori. In other words, the number of actions to be performed in a given time span is a parameter that has to be optimized itself. The number of genes contained in one chromosome is hence not fixed, and the algorithm allows the chromosomes to change their length while populations evolve. A genetic planner with variable length chromosomes has hence been implemented for ASOPS. It is to be noted that each chromosome formed this way represents a schedule (feasible or not) for the spacecraft.

Fitness Function One of the most important item of a Genetic Algorithm is its fitness function. It is used to select the "fittest" individuals of the population, to be mated and propagate the genetic material to the next generation. The fitness function created for ASOPS gets the chromosome (schedule) as input and simulates the behaviour of the spacecraft when executing that schedule. The value of the fitness function is obtained by considering multiple factors as: number of actions scheduled, constraints on energy and data storage, variety of actions scheduled, and temporal constraints. Each of these factor is independently computed during the simulation of the schedule and the final value of the fitness function is a weighted sum of the different factors. This approach allows flexibility by giving the possibility to easily change the weights of the different factors or to add a completely new factor if desired. It is also important to note that the outcome of the Genetic Algorithm is highly dependent on the weights set for the fitness function and in general a "calibration" of the same is required, according to the desired priorities.

Selection The selection technique used is the roulette selection. With this technique, the fitness function is used to associate a probability to each individual in the population to be selected for mating. In this process, there is the chance that some less fit solutions survive the selection, which is beneficial for the Genetic Algorithm, since weak solutions can still have good features that are this way kept in the population.

Genetic Operators Genetic operators characterize the way populations evolve by defining the way parents generate children (crossover operator) and how chromosomes can mutate (mutation operators). The crossover operator is a one point crossover operator: after the selection of the two parents to mate, a random crossover point is selected. The two parents are cut in the selected crossover point and the children are created by joining the beginning of one parent with the end of the other and vice-versa.
Additionally to the crossover operator, several mutation operators have been implemented in the genetic planner:

- **simple mutation**: this operator selects a random number of genes in the chromosome and mutates the execution times of the selected genes.
- **swap mutation**: this operator randomly selects 2 genes in the chromosome and swaps their execution times.
- **growth mutation**: this operator create randomly a new gene and adds it into the chromosome. The mutated chromosome will be hence longer than the original.
- **shrink mutation**: this operator randomly selects a gene from a chromosome and removes it. The mutated chromosome will be shorter than the original.

Algorithm Algorithm 1 describes the steps performed during the execution of the Genetic Algorithm implemented for ASOPS. The algorithm is an iterative process through which populations are evolved and more "fit" individuals are generated. Each operator mentioned in the algorithm loop is described in the previous paragraphs.

Algorithm 1 Genetic Algorithm

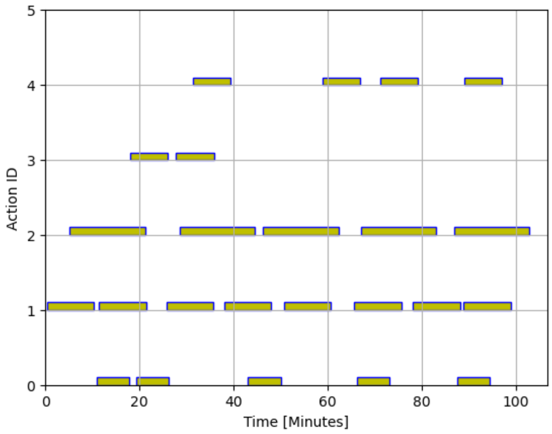
Initialize a random population of *pop_size* individuals(chromosomes).
while $iter \leq MaxIter$ **do**
  Apply *roulette selection* to create a poll of parents.
  Apply *crossover operator* to couples of parents.
  Add resulting children to population.
  Apply *mutation operators* to the population.
  Add mutated chromosomes to the population.
  Keep the fittest *pop_size* chromosomes.
  Update population.
end while

3.2 Simulation Environment

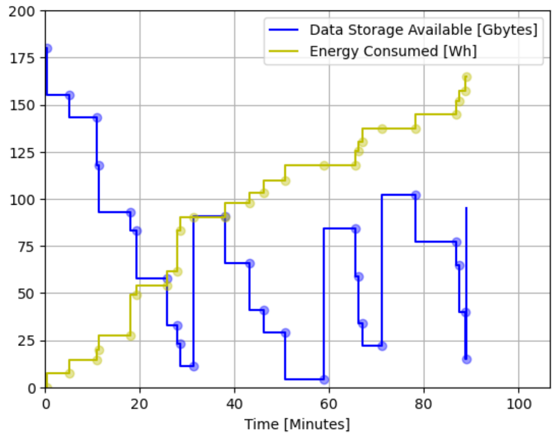
A scenario for a spacecraft carrying four different payloads has been used to test the performance of the Genetic Algorithm in the scheduling problem. Each payload has been simplified and modeled in terms of the power it draws during operations, the data it generates, and the duration of its operation. Furthermore, payload data downlink operations are also considered in the scheduling problem. The values of power consumption, data generation, and runtime of payload operations are taken from Valispace [25] and reflect payloads which will be part of the Athene-1 spacecraft. Valispace is a data management tool used as database for SeRANIS. The database of the algorithm of the possible actions to be scheduled is shown in Table 2. Since the goal is to test the Genetic Algorithm itself, it is to be noted that no complex functions are implemented for the way the payloads draw resources from the platform. Available resources on-board are simply updated when a scheduled action is executed. In line with Athene-1 performance, it is assumed that the energy available for payload operations is 180Wh per orbit and the schedule is generated for one orbit. Limits on the minimum energy and storage allowed by the spacecraft are also set in the simulation, to avoid completely

Table 2: Actions scheduled with Genetic planner.

Action	Action ID	Power Consumption [W]	Data Generated [GByte]	Duration [min]
Payload 1 Operation	0	45	25	7
Payload 2 Operation	1	46	25	10
Payload 3 Operation	2	26	12	16
Payload 4 Operation	3	160	10	8
Data Downlink	4	0	-80	8



(a) Schedule computed with Genetic Planner.



(b) Resources during execution of schedule.

Figure 4: Schedule computed with Genetic Planner and its resource utilization.

empty batteries and completely full storage. Also, the orbital environment and mission constraints are not considered yet. For example, the data downlink does not take into account ground station visibility and availability.

3.3 First Results

The schedule computed with the Genetic planner is visualized in Figure 4a. In Figure 4b, each dot represents the start of the scheduled action as displayed in Figure 4a. Each action has an effect on the on-board resources. As mentioned in the Chapter 3.2, in the current model resources on-board the spacecraft are updated each time a scheduled action starts, hence the step plot. The algorithm tries to schedule as many operations as possible in order to maximize the resource utilization, still respecting all the constraints imposed. It can be seen how every time the data storage available approaches its lower limit, the algorithm schedules a "Data Downlink" action and storage is freed to allow for more payload operations. The available energy utilization is also maximized by the calculated schedule, still respecting the limit on the minimum energy allowed on-board that is set to 10Wh.

3.4 Further Development

As highlighted in 3.2, the simulation environment is still basic and additions are necessary in order to model the complex real system. Detailed models for different payloads and subsystems of the platform are under development and will be integrated in the optimization process together with models for environment simulations. Other optimization methods as for example reinforcement learning are currently under investigation and will be compared with the performances of the Genetic Algorithm. With respect to the Genetic Algorithm, further investigation will be conducted in order to include additional constraints as for example ground station visibility, inter-dependency of payloads operation, or attitude of the satellite. Additionally, the number of possible operations to be scheduled will be increased. Also, given the many different parameters of a Genetic Algorithm, there is still room for improvement of the fitness function, where more parameters can be included; more complex crossover operators are also under investigation as well as new mutation operators. Additionally, methods to mitigate the risk of getting stuck in local optima like multipopulations and population reset will be investigated.

4 Conclusion

This paper presented the concept for the Autonomous Space Operations Planner and Scheduler (ASOPS). In a scenario where autonomy on-board spacecraft is sought after, ASOPS introduces a new concept to increase the level of autonomy of the spacecraft and increase the efficiency of its operations by optimizing the on-board resources utilization. The Seamless Radio Access Network for Internet of Space (SeRANIS) project represents a unique opportunity to assess ASOPS in space, in a complex satellite mission flying a large multitude of payloads. Nevertheless, ASOPS is meant to be not mission specific, but rather easily adaptable to any satellite mission thanks to its modularity and well defined interfaces. Together with on-board autonomy, the optimization of the schedule is the crucial objective of ASOPS. First results with Genetic Algorithms have shown them to be potentially suitable as optimization algorithm, but additional methods need to be investigated before a final assessment.

Acknowledgements

This work was supported by the dtec.bw—Digitalization and Technology Research Center of the Bundeswehr through the Project SeRANIS under Grant 150009910.

References

[1] H. A. Abbass, J. Scholz, and D. J. Reid, *Foundations of trusted autonomy*. Springer Nature, 2018.

[2] Z. für Digitalisierungs-und Technologieforschung der Bundeswehr, "Seranis a multifunctional satellite laboratory," https://dtecbw.de, 2022.

[3] A. Kinzel, J. Bachmann, R. Jaiswal, M. Karnal, E. R. Novo, F. Porcelli, A. Schmidt, R. Schwarz, C. Hofmann, R. Förstner, and A. Knopp, "Seamless radio access network for internet of space (seranis): New space mission for research, development, and in-orbit demonstration of cutting-edge technologies." Proceeding of the 73rd International Astronautical Congress (IAC) Paris, France, 2022.

[4] A. Jònsson, R. A. Morris, and L. Pedersen, "Autonomy in space: Current capabilities and future challenges," *AI Magazine*, vol. 28, 2007.

[5] W. Truszkowski, H. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt, *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*. Springer, 2009.

[6] ESA-ESTEC, "Ecss-e-st-70-11c – space segment operability," 2008. [Online]. Available: https://ecss.nl/standard/ecss-e-st-70-11c-space-segment-operability/

[7] F. Xhafa and A. W. Ip, "Optimisation problems and resolution methods in satellite scheduling and space-craft operation: a survey," *Enterprise Information Systems*, vol. 15, no. 8, pp. 1022–1045, mar 2019.

[8] S. Chien, D. Tran, G. Rabideau, S. Schaffer, D. Mandl, and S. Frye, "Timeline-Based Space Operations Scheduling with External Constraints," in *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS 2010)*, 2010.

[9] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, sep 2002.

[10] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "A comparison of techniques for scheduling earth observing satellites." *The Sixteenth Innovative Applications of Artifical IntelligenceConference (IAAAI04), San Jose, California, USA, July 25-29 2004.*, 2004.

[11] A. Sarkheyli, A. Bagheri, B. Ghorbani-Vaghei, and R. Askari-Moghadam, "Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 287–295, aug 2013.

[12] A. Weber, S. Fasoulas, and K. Wolf, "Generating optimised satellite payload operation schedules with evolutionary algorithms," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, may 2009.

[13] S. J. Russell, P. Norvig, S. Russell, and Russell, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.

[14] B. Sun, W. Wang, X. Xie, and Q. Qin, "Satellite mission scheduling based on genetic algorithm," *Kybernetes*, vol. 39, no. 8, pp. 1255–1261, aug 2010.

[15] S. woo Baek, S. mi Han, K. rae Cho, D. woo Lee, J. sik Yang, P. M. Bainum, and H. dong Kim, "Development of a scheduling algorithm and GUI for autonomous satellite missions," *Acta Astronautica*, vol. 68, no. 7-8, pp. 1396–1402, apr 2011.

[16] F. Xhafa, J. Sun, A. Barolli, M. Takizawa, and K. Uchida, "Evaluation of genetic algorithms for single ground station scheduling problem," in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*. IEEE, mar 2012.

[17] F. Sun and H. Wang, "Research on detection mission scheduling strategy for the LEO constellation to multiple targets," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 18, no. 2, pp. 87–103, jul 2017.

[18] P. Han, Y. Guo, C. Li, H. Zhi, and Y. Lv, "Geo satellites on-orbit repairing mission planning with missiondeadline constraint using a large neighborhood search-geneticalgorithm," 2021.

[19] E. V. Ntagiou, C. Iacopino, N. Policella, R. Armellin, and A. Donati, "Ant-based mission planning: Two examples," in *2018 SpaceOps Conference*. American Institute of Aeronautics and Astronautics, may 2018.

[20] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," Oct. 2017.

[21] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ser. NIPS'99. Cambridge, MA, USA: MIT Press, 1999, p. 1057–1063.

[22] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," 2016.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Jul. 2017.

[24] A. Hadj-Salah, R. Verdier, C. Caron, M. Picard, and M. Capelle, "Schedule earth observation satellites with deep reinforcement learning," 2019.

[25] "Valispace," Jan 2023. [Online]. Available: https://www.valispace.com/