Multiple Agile Satellite Scheduling

Offline-Online Multiple Agile Satellite Scheduling using Learning and Evolutionary Optimization

By

ABHIJIT CHATTERJEE, M.Sc. (Eng)

A Thesis Submitted to the School of Graduate Studies
in the Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

McMaster University
Hamilton, Ontario

Doctor of Philosophy (2023)
Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada

TITLE: Offline-Online Multiple Agile Satellite Scheduling using Learning and Evolutionary Optimization

AUTHOR:
Abhijit Chatterjee

SUPERVISOR:
Dr. Ratnasingham Tharmarasa
Assistant Professor, Department of Electrical and Computer Engineering,
McMaster University, ON, Canada

NUMBER OF PAGES: xiv, 150

To my beloved Proma

Abstract

The recent generation of Agile Earth Observation Satellite (AEOS) has emerged to be highly effective due to its increased attitude maneuvering capabilities. However, due to these increased degrees of freedom in maneuverability, the scheduling problem has become increasingly difficult than its non-agile predecessors. The AEOS scheduling problem consists of finding an optimal assignment of user-requested imaging tasks to the respective AEOSs in their orbits by satisfying the operational resource constraints in a specified time frame. Some of these tasks might require imaging the same area of interest (AOI) multiple times, while in some tasks, the AOIs are too large for the AEOS to image in a single attempt. Some tasks might even arise while the AEOSs are preoccupied with existing tasks.

This thesis focuses on formulating the AEOS scheduling models where onboard energy and memory constraints while operating and the task specifications are diverse. A mixed-integer non-linear scheduling problem with a reward factor has been considered in order to handle multiple scan requirements for a task. Although initially, it is assumed that the AOIs are small, this work is extended to a three-stage optimization framework to handle the segmentation of large AOIs into smaller regions that can be imaged in a single scan. The uncertainty regarding scan failure is handled through a Markov Decision Process (MDP). These two proposed methods have significant benefits when tasks are available to schedule prior to the mission. However, they lack the flexibility to accommodate newly arrived tasks during the mission. When multiple new tasks arrive during the mission, predictive scheduling based on learning historical data of task arrivals is proposed,

which can schedule tasks in an online manner faster than complete rescheduling and minimize disruption from the original schedule. Evolutionary optimization-based solution methodologies are proposed to solve these models and are validated with simulations.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. R. Tharmarasa for being an excellent guide with his scholarly insights and extensive expertise. His immense patience and constructive feedback have greatly helped me improve my research ability. His kind supportive nature and caring attitude have kept me sane during the global pandemic of 2019-21. I would also like to thank my supervisory committee Prof. T. Kirubarajan and Prof. A. Jeremic for all their wonderful insights to enrich my research manifold. I am indebted to the Department of Electrical and Computer Engineering, McMaster University for providing me with the funding, resources, and such an incredible experience during my Ph.D.

I extend my acknowledgment to all my lab mates at ETFLab, Sarojini, Prabhanjan, Anbang, Ming, Dipayan, Aranee, Honghao, and Jing for all the support.

This work would have been impossible without the support of my loving family and friends. I can't express enough gratitude to Debjani aunty and Atanu uncle for their love and guided advice throughout my Ph.D. journey. Thank you Debmalya for making Canada feel a little close to home. I thank my parents, my uncle and aunt, my brother and sister-in-law for all their support and encouragement.

Lastly, the most important part of my life, I thank Dr. Aishwaryaprajna for being the most forbearing and caring partner ever through all those late-night discussions and tolerating every tantrum I made.

Contents

List of Figures

List of Tables

Declaration of Academic Achievement

I, Abhijit Chatterjee, declare that this thesis titled, **Offline-Online Multiple Agile Satellite Scheduling using Learning and Evolutionary Optimization**, and works presented in it are my own. I confirm that analytical and computational work carried out solely by Abhijit Chatterjee, herein referred to as "the author", with advice and guidance provided by the academic supervisor Dr. R. Tharmarasa. Information that is presented from outside sources and has been used towards analysis or discussion, has been cited when appropriate, all other materials are the sole work of the author.

Chapter 1

Introduction

Earth-observing satellites (EOSs) provide a unique perspective from space orbits for information acquisition required in surveillance and tracking missions, which is often beyond the capability of sensors placed on the Earth's surface. Due to the advantage of the high altitude of EOS, they have better access to information arising from much larger area of interest (AOI) and difficult to reach AOIs due to hostile or remote environmental conditions. These enable EOSs to perform real-world applications, including weather forecasting, monitoring agricultural attributes like crop health, moisture detection, and insect infestations, climate disasters like forest fires, cyclones, and tsunamis, as well as, tracking land and marine traffic, often required in defence applications.

These observation tasks might often require consistent, long-term, or time-sensitive imaging of the AOIs. Since EOSs have predefined orbits, achieving the success of the observation tasks within a specified time frame is a challenge to assign EOSs to respective tasks. As multiple EOSs may be involved in a mission of performing a series of tasks, the assignment of task schedules to EOSs is necessary for efficiency. This assignment requires considering the resource constraints of the respective EOSs while planning the task schedule. This problem of satisfying multiple constraints simultaneously is computationally challenging and NP-hard in nature (Hall and Magazine; 1994; Lemaître et al.; 2002). Uncertainty arising from cloud coverage, sensor malfunctions and other unexpected events increases the complexity of the EOS scheduling problem manifold.

The workflow of EOS systems at the ground stations generally consists of – collecting observation tasks from the users, planning the optimal task schedule of the EOSs, up-linking the schedule to the EOSs, down-linking the obtained image data from the EOSs and processing the information before sending it to the users. The focus of this thesis lies in analyzing and modeling the EOS scheduling problem for the user-requested observation tasks, handling uncertainties, identifying appropriate methods for scheduling the tasks and assigning the EOSs to the tasks, and considering variants in task specifications while satisfying the necessary resource constraints of the EOSs.

1.1 Agile Satellites vs. Non-Agile Satellites

The emerging generation of EOSs, Agile EOSs (AEOSs), has significantly widened the capabilities of satellite observations than its non-agile predecessors. The inclusion of the Attitude Determination and Control System (ADCS) has provided EOSs with additional maneuverability in terms of the orientation (attitude) of the sensors mounted in the satellites.

Non-Agile EOSs (N-AEOSs) like SPOT systems of EOS[1] use a single degree of freedom for sensors to manoeuvre in the *roll* axis for image acquisition. The roll axis is parallel to the orbital trajectory of the satellite. The N-AEOSs can only perform imaging tasks when they are positioned exactly above the AOIs. This makes the visible time window (VTW) of an AOI extremely limited. In conflicting situations arising from overlapping VTWs, the problem reduces to a selection problem rather than a true scheduling problem (Lemaître et al.; 2002).

On the contrary, AEOS like *GeoEye-1*[2], *WorldView*[3], *SuperView-1*[4], *Pleiades*[5] and *EOS-04*[6] have three degrees of freedom in *roll*, *pitch* and *yaw* axes. The additional agility of AEOS provides significantly more opportunities for AOI imaging. The maneuverability in pitch and yaw directions omit the restrictions posed in N-AEOS systems.

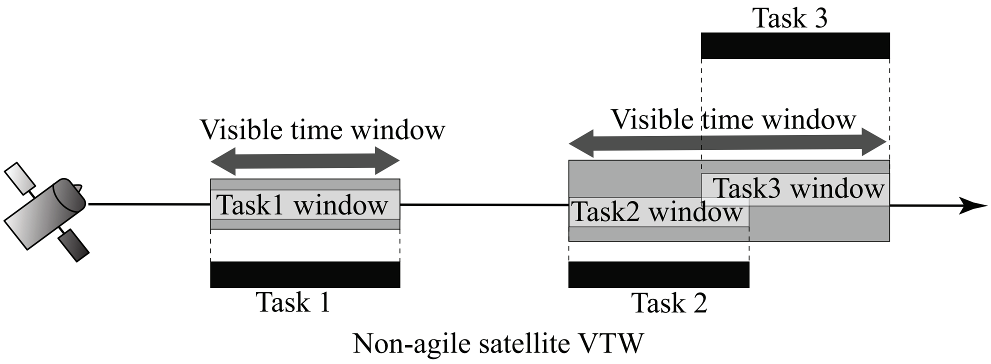


Figure 1.1: Visible time window for Non-Agile satellites

The figures 1.1 and 1.2 illustrate the difference in the VTWs arising from the varied degrees of freedom in N-AEOS and AEOS. Since the starting time of image

[1] https://spot.cnes.fr/en/SPOT/index.htm
[2] https://earth.esa.int/eogateway/missions/geoeye-1
[3] https://worldview.earthdata.nasa.gov
[4] https://eos.com/find-satellite/superview-1/
[5] https://earth.esa.int/eogateway/missions/pleiades
[6] https://www.isro.gov.in/EarthObservationSatellites.html

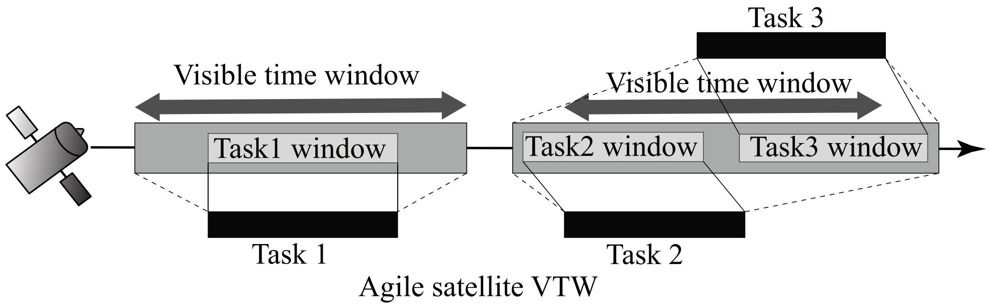


Figure 1.2: Visible time window for Agile satellites

acquisition for AEOS can be more flexible than N-AEOS, it provides infinitely more imaging possibilities to the AEOS. In situations where the N-AEOSs fail to perform imaging tasks due to overlapping of AOIs along the timeline, the AEOSs can perform those imaging tasks by using the advantage of the higher degrees of freedom. However, this makes the scheduling problem for AEOSs significantly harder due to the substantially larger search space.

1.2 Resource Constraints in AEOS Scheduling Problem

In this thesis, the considered AEOS scheduling problems have operational constraints regarding the satellite resource specifications like energy consumption and memory requirements. An optimal assignment of a sequence of tasks to respective satellites by satisfying the resource constraints is necessary to complete the user requests in the specified time frame (Hall and Magazine; 1994).

The EOSs do not require any energy to revolve around the Earth. The equilibrium of their centrifugal force and gravity helps the satellites to revolve. However, these satellites require energy to image using the sensors, transition from one task to another while orbiting by rotating the sensors along with the roll and pitch angles, and transfer data to and from the ground stations. The amount of energy needed for each action is different; thus, the discharging rate varies for each action. The satellites have solar sensors that automatically direct the solar panels toward the Sun so that the satellite's onboard batteries can charge in with the maximum possible charging rate. Charging and discharging constraints have been used for AEOS scheduling (Liu et al.; 2017; Han et al.; 2022), N-AEOS scheduling (Baek et al.; 2011) and nano-satellite swarms (Pang et al.; 2015).

The task allocation and schedule of the tasks are up-linked to the satellites by ground stations. After completing the tasks, the satellites down-link the data to ground stations. In order to store the data until it is down-linked, a limited resource of on-board memory is used. Hence, periodically, the satellites need to transfer the collected data to the ground stations in order to free up the memory storage. A study on AEOS scheduling (Liu et al.; 2017) has mentioned the on-board memory constraint in their model formulation without including it in the algorithm and simulation results. In (Peng et al.; 2020; Han et al.; 2022), it is assumed that the satellites have sufficient memory to complete all the tasks in each orbit. A realistic constraint, where the satellites need to clear the memory by down-linking to the ground station in order to complete all the tasks, is considered in this thesis.

1.3 User-requested Tasks in AEOS Scheduling Problem

The AOIs corresponding to the user-requested imaging tasks might be spread across the Earth's surface in a scattered manner. The visible window (VW) of the AEOS depends on its location and is based on the on-board sensor specifications (maximum roll and pitch angles). The task AOIs can only be scanned when they belong to the VW of the AEOS.

In literature, a reward-based optimization problem is often formulated, where maximizing the reward provides an optimal task schedule for the satellites. The reward after the successful completion of a task is also provided by the user. Depending on the nature of a task, single or multiple scans might be needed in order to complete that task. A reward function based on the quality of the image is used in (Wolfe and Sorensen; 2000). A step-function-based reward factor is used to deal with the multiple scan requirement for a task for N-AEOS scheduling in (Tharmarasa et al.; 2019). In this thesis, a similar reward factor is considered in Chapter 2 for AEOS scheduling to calculate the total reward accumulated for each task at the end of the mission horizon.

The task AOIs for the AEOSs are usually classified as spot and polygon targets. Spot targets or polygon targets with smaller AOIs have a limited area to scan, which can be completed by one pass according to the observation scope of the AEOS sensor, while larger polygon targets may need to be captured by multiple passes of multiple AEOSs in a cooperative manner. When a request to scan a large region is received, the request either involves the region to be scanned in a geometrically consistent manner or involves scanning the whole region within a specified time window.

A large portion of the literature focuses on spot targets, comparatively smaller polygon targets or sub-tasks with smaller regions originating from large polygon targets (Habet et al.; 2010; Wang et al.; 2020). AEOS scheduling for spot target-based scanning tasks is relatively easier as the length or the orientation of the task length does not need to be considered in the problem (Renjie et al.; 2008). Based on the task requests from the users, larger AOI scanning is an extremely relevant and realistic problem to address, as in applications like monitoring forest fires, crop growth for large regions, or iceberg melting rates. When the AOIs are too large for AEOSs to scan in one go, the complexity of scheduling multiple AEOSs in their multiple revolutions to perform multiple scans for each task increases manifold (Niu et al.; 2018).

The user-requested tasks can be further classified based on the time of task arrival to the scheduler. Some tasks are daily imaging tasks, which are known to the scheduler prior to the start of the mission. These tasks can be scheduled before the start of the mission as an offline optimization problem.

Some task arrivals can occur during an ongoing mission. Recent literature on AEOS schedule updations considers online rescheduling when real-time information is available (Chu et al.; 2017). However, in dynamic environments, complete rescheduling with real-time information requires a huge computational load and valuable time and will also create havoc by disrupting the original schedule.

1.4 Main Contributions of the Thesis

In compliance with the terms and regulations of McMaster University, this thesis has been written in *sandwich thesis* format by assembling three articles. These articles represent the independent research performed by the author of this thesis, Abhijit Chatterjee under the supervision of Dr. Ratnasingham Tharmarasa. The contributions of the thesis are as follows,

- Formulation of realistic AEOS scheduling models with operational constraints related to energy and memory for performing user-requested tasks with varied specifications (in Chapters 2, 3 and 4)

- Consideration of probabilistic uncertainty in task completion and multiple scan requirements to achieve full reward for a task (in Chapter 2)

- Minimization of scan overlaps for large task AOIs by providing novel area segmentation strategy (in Chapter 3)

- MDP-based uncertainty handling for task failures (in Chapter 3)

- Learning-based predictive rescheduling strategy to accommodate new task arrivals during the mission with complete rescheduling (in Chapter 4)

- Development of algorithmic frameworks for solving AEOS scheduling problems in an offline-online manner (in Chapters 2, 3 and 4)

1.5 Related Publications

1. **Chatterjee, Abhijit** and Tharmarasa, Ratnasingham (2022) Reward factor-based multiple agile satellites scheduling with energy and memory constraints, *IEEE Transactions on Aerospace and Electronic Systems* **58**(4): 3090–3103.

2. **Chatterjee, Abhijit** and Tharmarasa, Ratnasingham (2023) Multi-stage optimization framework of satellite scheduling for large areas of interest, *First revision under review, Advances in Space Research, Elsevier.*

3. **Chatterjee, Abhijit** and Tharmarasa, Ratnasingham (2023) Learning-based predictive scheduling for multiple agile satellites with new task arrivals during mission, *Submitted to IEEE Transactions on Aerospace and Electronic Systems.*

Bibliography

Baek, S.-w., Han, S.-m., Cho, K.-r., Lee, D.-w., Yang, J.-s., Bainum, P. M. and Kim, H.-d. (2011). Development of a scheduling algorithm and GUI for autonomous satellite missions, *Acta Astronautica* **68**(7-8): 1396–1402.

Chu, X., Chen, Y. and Tan, Y. (2017). An anytime branch and bound algorithm for agile earth observation satellite on-board scheduling, *Advances in Space Research* **60**(9): 2077–2090.

Habet, D., Vasquez, M. and Vimont, Y. (2010). Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite, *Computational optimization and applications* **47**: 307–333.

Hall, N. G. and Magazine, M. J. (1994). Maximizing the value of a space mission, *European journal of operational research* **78**(2): 224–241.

Han, C., Gu, Y., Wu, G. and Wang, X. (2022). Simulated annealing-based heuristic for multiple agile satellites scheduling under cloud coverage uncertainty, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(5): 2863–2874.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M. and Bataille, N. (2002). Selecting and scheduling observations of agile satellites, *Aerospace Science and Technology* **6**(5): 367–381.

Liu, X., Laporte, G., Chen, Y. and He, R. (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time, *Computers & Operations Research* **86**: 41–53.

Niu, X., Tang, H. and Wu, L. (2018). Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm, *International journal of disaster risk reduction* **28**: 813–825.

Pang, C. K., Kumar, A., Goh, C. H. and Le, C. V. (2015). Nano-satellite swarm for SAR applications: Design and robust scheduling, *IEEE Transactions on Aerospace and Electronic Systems* **51**(2): 853–865.

Peng, G., Song, G., Xing, L., Gunawan, A. and Vansteenwegen, P. (2020). An exact algorithm for agile earth observation satellite scheduling with time-dependent profits, *Computers & Operations Research* p. 104946.

Renjie, H., Baocun, B., Yingwu, C. and Yuejin, T. (2008). Multi-satellite mission planning for environmental and disaster monitoring satellite system, *SpaceOps 2008 conference*, p. 3488.

Tharmarasa, R., Kirubarajan, T., Berger, J. and Florea, M. C. (2019). Mixed open-and-closed loop satellite task planning, *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–8.

Wang, X., Wu, G., Xing, L. and Pedrycz, W. (2020). Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions, *IEEE Systems Journal* .

Wolfe, W. J. and Sorensen, S. E. (2000). Three scheduling algorithms applied to the earth observing systems domain, *Management Science* **46**(1): 148–166.

Chapter 2

Reward Factor-Based Multiple Agile Satellites Scheduling With Energy and Memory Constraints

The content of this chapter is a publication in IEEE Transactions on Aerospace and Electronic Systems under the following citation:

Chatterjee, Abhijit and Tharmarasa, Ratnasingham (2022) Reward factor-based multiple agile satellites scheduling with energy and memory constraints, *IEEE Transactions on Aerospace and Electronic Systems* **58**(4): 3090–3103.

Reward Factor-Based Multiple Agile Satellites Scheduling with Energy and Memory Constraints

Abstract

Earth Observing Satellites (EOS) orbit around the earth to perform observation tasks specified by users. The additional maneuverability resulting from higher degrees of freedom than Non-Agile EOS (N-AEOS), provides Agile EOS (AEOS) a significantly larger Visible Time Window (VTW) to complete the tasks. As a consequence, the task scheduling for AEOS is much more computationally complex than N-AEOS. In this paper, a mixed-integer non-linear optimization problem is formulated to find the optimal task allocations for a realistic AEOS scheduling problem. The satellite resources like energy and memory constraints are considered in this problem. A reward factor is used to address the requirement of multiple scans in order to complete a task. To incorporate the uncertainty of successful scans due to external factors, such as cloud coverage, a probability factor is also taken into consideration. An elitist mixed coded genetic algorithm-based satellite scheduling (EMCGA-SS) algorithm is proposed to solve the formulated problem. EMCGA-SS is extended to elitist mixed coded hybrid genetic algorithm-based satellite scheduling (EMCHGA-SS) by combining a hill-climber mechanism in order to have better initialization. Experimental results to illustrate the performance of the algorithms and a comparison with some widely used methodologies

are also presented.

Keywords: *agile satellite scheduling, reward factor, energy and memory constraints, mixed-integer non-linear programming, genetic algorithm*

2.1 Introduction

Earth Observing Satellites (EOS) are equipped with various sensors to perform specific tasks such as weather predictions, surveillance and tracking by imaging the earth from space. Based on job requests from users, satellites need to perform an array of tasks. These task locations might be spread across a large region of interest in a scattered manner. An EOS revolves around the earth on its orbit, exposing segments of the earth surface to the satellite sensor depending on the location of the satellite in the orbit. Satellites might need to observe a region from different angles with respect to its orbit in order to perform a task. Possible time intervals when the satellite can observe the tasks are defined as the Visible Time Window (VTW). Satellite scheduling involves optimal allocation of the tasks to the satellites and optimal transition sequence of tasks by satisfying the resource constraints. Scheduling of task sequences optimally is an important aspect of satellite performance, as well as fulfillment of the job request.

Agile EOS (AEOS) such as *GeoEye - 1, WorldView, SuperView - 1* and *Pleiades* have three degrees of freedom – roll, pitch and yaw (Lemaître et al.; 2000). Whereas, Non-agile EOS (N-AEOS) have only one degree of freedom. Due to the extra degrees of freedom, an AEOS has a larger VTW to perform a task with respect to a N-AEOS, as shown in Figure 1.1 and Figure 1.2. This higher maneuverability provides AEOS significant advantages over N-AEOS. AEOS can start performing a task at any time within the larger VTW due to the higher number of possible start-times. This additional maneuverability allows AEOS to have more transition possibilities between tasks than N-AEOS. AEOS has a bigger coverage area than N-AEOS in a single orbit as well, due to the increased rotational ability. Consequently, scheduling an AEOS becomes significantly more difficult due to a larger task allocation search space. Since an AEOS scheduling problem is NP-hard (Lemaître et al.; 2002), the computation time needed to find optimality will grow exponentially with respect to the problem instance.

In this work, a realistic AEOS scheduling problem is studied, where satellite resource specifications like energy and memory are taken into consideration. The satellites require energy for performing the tasks and transitioning from one task to another while orbiting. The satellites have solar sensors which automatically direct the solar panels towards the sun so that, the satellite's on-board batteries can charge in with the maximum possible charging rate. Charging and discharging constraints have been used for N-AEOS scheduling (Baek et al.; 2011) and nano-satellite swarms (Pang et al.; 2015).

The task allocation and schedule of the tasks are up-linked to the satellites by ground stations. After completing the tasks, the satellites down-link the data to ground stations. In order to store the data until it is down-linked, a limited resource of on-board memory is used. A recent study (Liu et al.; 2017) has mentioned the on-board memory constraint in their model formulation without including it in the algorithm and simulation results. In (Peng et al.; 2020), it is assumed that the satellites have sufficient memory to complete all the tasks in each orbit. A more realistic formulation, where the satellites need to clear the memory by downlinking to the ground station in order to complete all the tasks, is considered in this current study.

In literature, a reward-based optimization problem is often formulated, where maximizing the reward provides an optimal task schedule of the satellites. A reward function based on the quality of the image is used in (Wolfe and Sorensen; 2000). A ten-level scale for image quality-based reward for each task has been introduced in (Liu et al.; 2017). The observation angle while performing a task depends on the location of the satellite in its trajectory and the location of the task. Due to increased VTW, AEOS can observe a task from a larger observation angle than N-AEOS. However, the quality of a scan might depend on the observation angle. Based on the start-time of the task performance in the VTW, the observation angle is calculated. In (Peng et al.; 2020), a reward function depending on the start-time of the task performance is considered. A reward function based on the observation angle is considered in the problem formulation of the present work.

Depending on the nature of a task, multiple scans might be needed in order to complete that task. A step-function-based reward factor is used to deal with the multiple scan requirement for a task for N-AEOS scheduling in (Tharmarasa, Kirubarajan, Berger and Florea; 2019). In this paper, a similar reward factor is considered for AEOS scheduling to calculate the total reward accumulated for each task at the end of the mission horizon.

In reality, interruption due to cloud coverage is a very common occurrence while performing a task by a satellite. A real-life data set of *Landsat-7* showed 35% of the captured images were covered with cloud. (Ju and Roy; 2008). Considering the present and future weather conditions, a quality-based problem is formulated in (Liao and Yang; 2007). In a similar manner, (Zhai et al.; 2015) also considered that no reward would be allocated if there is any uncertainty due to external factors in the satellite performance. In (Lin et al.; 2012), an image processing-based cloud removal technique is proposed. A budgeted uncertainty set-based formulation is considered in (Wang et al.; 2019). A probability-based success and failure of the operation are considered in (Tharmarasa, Chatterjee, Wang, Kirubarajan, Berger and Florea; 2019) for N-AEOS scheduling. A similar probability factor is included in the formulation of this work.

Early research (Lemaître et al.; 2000) on AEOS scheduling problem provided two approaches: constraint programming and local search for solving a simplified version of AEOS problem. Permutation-based search, coupled with constraint propagation over VTW, was found to work better than local optimization algorithms such as hill-climbing, simulated annealing, and squeaky wheel optimization, coupled with constraint propagation (Dilkina and Havens; 2005). A tabu-search heuristic is used to solve a scheduling problem in a given orbit in (Cordeau and Laporte; 2005). This work was extended for a practical operation of the *Pleiades* constellation for handling several satellites in multiple orbits over a given time horizon in (Bianchessi et al.; 2007). A deterministic constructive algorithm with a look ahead and backtracking capabilities was used to solve the planning and scheduling problem of *COSMO-SkyMed* satellite constellation (Bianchessi and Righini;

2008). A tabu-search algorithm was proposed for solving a constrained AEOS scheduling problem by maximizing the reward as well as minimizing the sum of transition durations (Habet et al.; 2010). A multi-objective AEOS management problem was solved with a biased random key genetic algorithm in (Tangpattanakul et al.; 2012). A metaheuristic, adaptive large neighbourhood search is shown to perform time-dependent transition time consideration for a single AEOS scheduling(Liu et al.; 2017). This work was extended to multiple AEOS scheduling in (He, Liu, Laporte, Chen and Chen; 2018). Single AEOS scheduling problem was also considered for redundant observation targets where each possible observation opportunity is considered a node in a complex network (Wang et al.; 2016). An anytime branch and bound on-board scheduling methodology was proposed on a bi-satellite cluster for target detection using low-resolution satellite and target recognition using a high-resolution agile satellite trailing behind the low-resolution satellite in (Chu et al.; 2017). A learning-based approach for AEOS scheduling is proposed in (Lu et al.; 2020) where random forest provided high-quality solutions to train an offline classifier on massive data from the ground station.

In general, heuristic search methods, including genetic algorithms perform efficiently in solving computationally complex problems like satellite scheduling. Here, an elitist genetic algorithm-based solution technique is proposed. The initial population of feasible candidate solutions is generated with and without the help of a hill-climber mechanism. Tabu-search and simulated annealing are compared with the performance of the proposed algorithms. The simulation results show that the hybrid hill-climber genetic algorithm yields better schedules than the other considered algorithms whereas, the proposed algorithm with random initialization provides a solution faster.

The next section discusses the assumptions considered in the realistic AEOS scheduling problem. Section 2.3 illustrates the objective function and the energy, memory and time constraints considered. Proposed heuristic-based solution methodologies and experimental results, along with a comparative study of the proposed algorithms with the widely used heuristic algorithms are described in sections 2.4 and 2.5, respectively. Concluding remarks and future scopes are added in section 2.6.

2.2 Problem Statement

In this paper, we consider that satellites need to perform some pre-defined observation tasks within the given time horizon. These user-specified tasks involve scanning particular geographical locations and obtaining rewards on completion. The imaging tasks may include identifying, tracking and surveillance of targets. The task locations are usually spots, i.e., of limited geographical dimension or widespread polygonal regions. The tasks are scattered on the whole of the earth's surface and are visible from different orbits of multiple satellites. The objective of this work is to schedule observation tasks optimally by allocating the available resources of agile satellites.

Satellites can scan the whole region of the task in one visit. Each task is associated with a specific reward. When a satellite successfully scans the region, a reward is assigned. The total reward is calculated by accumulating all the rewards gained by the satellites at the end of the mission horizon. The users are interested in maximizing the rewards by completing the requested tasks within a certain period of time. Finding the optimal sequence of scans to be performed by the AEOS to get the maximum reward possible is the primary objective of this work.

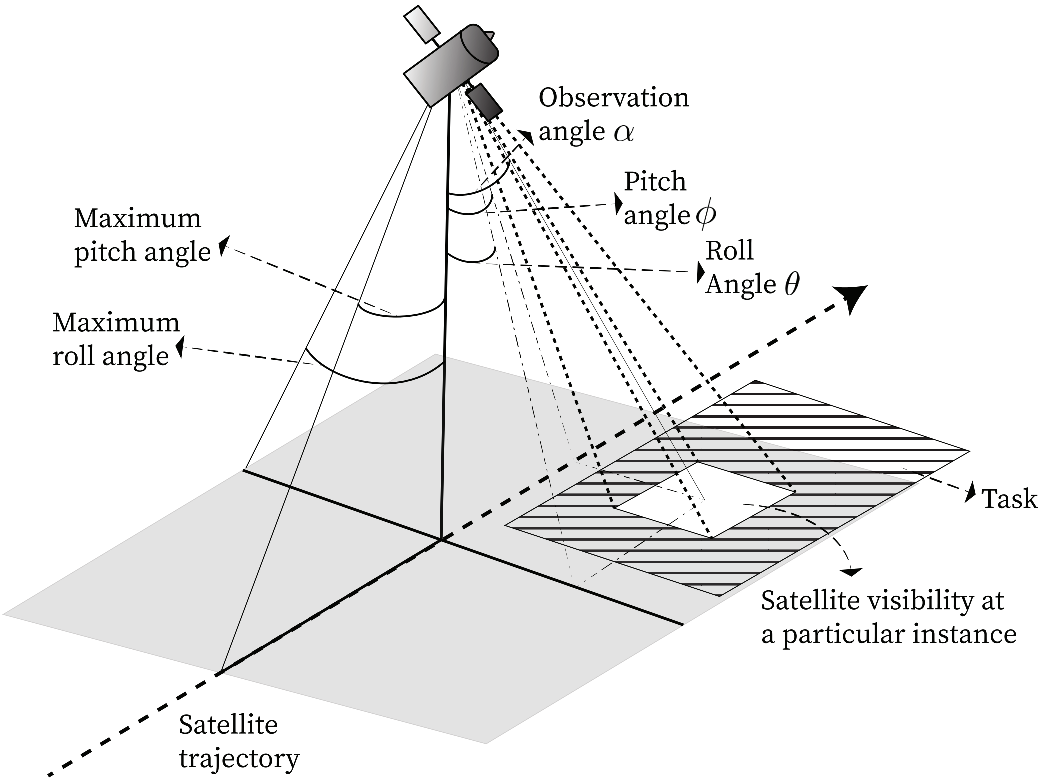


Figure 2.1: Observation angle of a satellite

In order to scan a particular task, the position of the satellites in their trajectory needs to be inside the VTW of the corresponding task. The VTW of a task for a satellite is calculated beforehand based on the geographic location of the task region, the orbit of the satellite and maximum roll and pitch angles of the satellite. Each possible start-time of a scan for the task within the VTW will have a different roll and pitch angle combination for every satellite on every orbit. The reward accumulated by a satellite after scanning a task may depend on the corresponding roll and pitch angles. As a result, it is important to schedule the scans in an

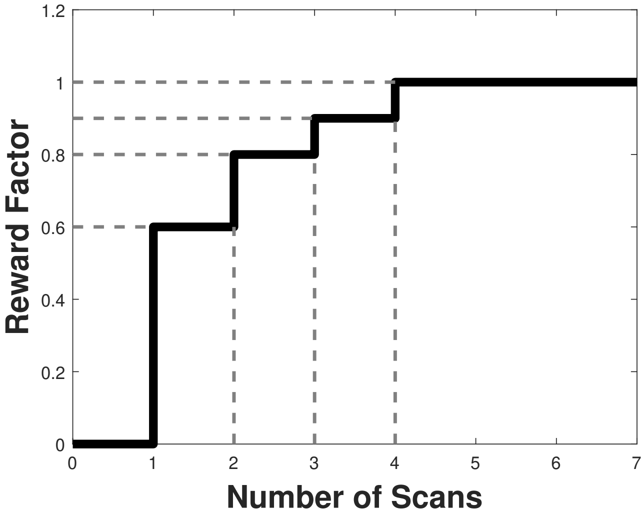


Figure 2.2: Reward factor vs number of scans

optimal way such that the reward is maximized based on the above-mentioned factors.

For several reasons like the presence of clouds, failure of equipment, and sensor distortions, there is always a chance for a failure of a scan in spite of the satellite being present in the VTW. Our scheduling model also incorporates the failure probability of a scan due to unprecedented reasons based on the outcome (success or failure) of each scan.

Multiple scans might be needed to complete a task and hence, to get the full reward for the task. A reward factor is considered to calculate the total reward achieved by multiple successful scans of a particular task. The scans can be performed by a single satellite or multiple satellites in different orbits. The reward acquired after several scans of a task is a step function of the number of successful scans. Based on the characteristics of the task, the user defines the step function.

The step sizes might not be equal for each scan. If $R(t)$ is the total reward achieved for completing task t, a reward factor $R^f_{(t,d)}(< 1)$ is introduced for task t on its d-th successful scan. Figure 2.2 shows an example reward factor function for a particular task with multiple scans the $R^f_{(t,d)}$ values are $(0.6, 0.2, 0.1, 0.1)$. If a task is visible by a satellite in a particular orbit, it is considered that the satellite will scan that task only once in that particular orbit.

The satellites do not require any energy to revolve around the Earth. The equilibrium of their centrifugal force and gravity helps the satellites to revolve. However, the satellites need energy to perform the necessary actions to complete the tasks. The actions include imaging using the sensors, rotating the sensors along with the roll and pitch angles and transferring data to and from the ground stations. The amount of energy needed for each action is different thus, the discharging rate varies for each action. Figure 2.3 shows an example of satellite charging and discharging of a satellite over multiple orbits. The locations of the ground stations are fixed and known. The satellites have limited memory capacity. Hence, periodically the satellites need to transfer the collected data to the ground stations to free up the memory space. A new task schedule is also transferred from the ground station to the satellite periodically. The optimal times of communication between the ground station and the satellites are needed to be calculated. The objective of this work is to assign tasks to satellites optimally in order to maximize the total reward over the time horizon while considering all the constraints.

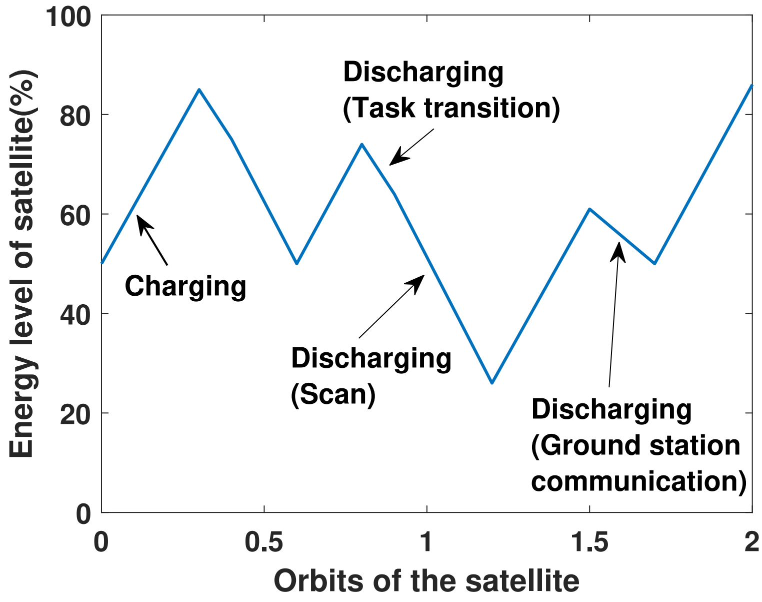


FIGURE 2.3: Charging and discharging of energy for satellites in each orbit

2.3 Problem Formulation

In this section, a non-linear mixed-integer programming problem is formulated such that the reward for the whole mission is maximized while satisfying all the constraints. Several constraints related to the satellite's resources are included. In this formulation, the start-time of each operation is considered as a decision variable, which is continuous in nature. The operations of a satellite include performing the scans, and data transfer to and from the ground stations. To determine whether a satellite is performing a particular scan on a specific orbit, a binary decision variable is also defined. The variables used for the formulation are described below.

Decision variables:

- $a_{(s,o,t)}$: Binary decision variable to denote satellite s performs task t in orbit o
- $T^{st}_{(s,o,t)}$: Start-time of scanning for task t by satellite s in orbit o
- $T^{sg}_{(s,o,g)}$: Start-time of data transfer from satellite s in orbit o to ground station g

Other variables:

- N^t: Number of tasks to complete
- N^s: Number of satellites
- N^{os}: Number of orbits in the trajectory of satellite s in the given time horizon
- D^{max}_t: Maximum number of successful scans needed to complete task t
- $\tau_{(s,o,t)}$: Time needed by satellite s for scanning task t in orbit o
- $T^{VTW_{st}}_{(s,o,t)}$: Start-time of the VTW for satellite s in orbit o for scanning task t
- $T^{VTW_{end}}_{(s,o,t)}$: End-time of the VTW for satellite s in orbit o for scanning task t
- $T^{VTW_{st}}_{(s,o,g)}$: Start-time of the VTW for satellite s in orbit o to transfer data to ground station g
- $T^{VTW_{end}}_{(s,o,g)}$: End-time of the VTW for satellite s in orbit o to transfer data to ground station g
- $R^f_{(t,d)}$: Reward factor for completing d^{th} scan of task t

- $R_{(s,\alpha,t)}$: Reward for scanning task t by satellite s with observation angle α (includes roll and pitch angles)
- $P_{(s,o,t)}$: Probability of success for particular scan of task t by satellite s in orbit o
- $P^{suc}_{(s,t,d)}$: Probability of getting d successful scans for task t by satellite s
- $e_{(s,o,t)}$: Energy required by satellite s to scan task t in orbit o
- e^{roll}_{s}: Energy required by satellite s to roll a unit angle
- e^{pitch}_{s}: Energy required by satellite s to pitch a unit angle
- τ^{roll}_{s}: Time required by satellite s to roll a unit angle
- τ^{pitch}_{s}: Time required by satellite s to pitch a unit angle
- $\theta^{t,\hat{t}}_{(s,o)}$: Roll angle needed for satellite s in orbit o to move from task t to task $\hat{t}$
- $\phi^{t,\hat{t}}_{(s,o)}$: Pitch angle needed for satellite s in orbit o to move from task t to task $\hat{t}$
- $e^{dl}_{(s,o)}$: Average energy required for a down-link by satellite s
- $e^{ul}_{(s,o)}$: Average energy required for an up-link by satellite s
- $m_{(s,t)}$: Memory usage of satellite s to scan task t
- m^{max}_{s}: Maximum memory capacity of satellite s

- τ_s^{mem}: Time needed to transfer data between satellite and ground station
- $N_{(s,g,\hat{g})}$: Number of scans performed between communication of satellite s to ground station g and the next ground station $\hat{g}$
- $C_{(s,\gamma)}$: Available charge of satellite s at time γ
- ch_s: Charging rate for satellite s

2.3.1 Objective function calculating accumulated reward

The objective function involves optimizing the reward accumulated by all the satellites during the mission by performing several successful scans of the tasks. The objective function is defined as follows:

$$\max \sum_{t=1}^{N^t} \sum_{d=1}^{D_t^{max}} R^f_{(t,d)} \rho_{(t,d)} \tag{2.1}$$

$$\text{where,} \qquad \rho_{(t,d)} = \sum_{s=1}^{N^s} \sum_{o=1}^{N^{os}} R_{(s,\alpha,t)} P^+_{(t,d,s,o)} \tag{2.2}$$

Here, the reward factor $R^f_{(t,d)}$ is included in the objective function to calculate reward for different scans of the same task and $\rho_{(d,t)}$ calculates the reward for d^{th} successful scan of task t. The reward $R_{(s,\alpha,t)}$ achieved by scanning might depend on the observation angle α of the task t from the satellite s. The relationship of reward with observation angle is specific to the task and defined by the user. In general, the reward is inversely proportional to observation angle. The angle α is calculated as a combination of the roll and pitch angle movements of the satellite depending on the position of the task. An example of the relationship is provided in the simulation in Section 2.5.

A scan can be unsuccessful due to several factors, like weather influences and sensor malfunctions. $P^{+}_{(t,d,s,o)}$ is defined as the probability of having d^{th} successful scan for task t by satellite s in orbit o. The probability of d^{th} successful scan be defined as:

$$P^{+}_{(t,d,s,o)} = a_{(s,o,t)} P_{(s,o,t)} P^{-}_{(t,(d-1),s,o)} \tag{2.3}$$

where, $P^{-}_{(t,(d-1),s,o)}$ is the probability of having $d-1$ successful scans before the start time ($T^{st}_{(s,o,t)}$) of the scan by satellite s in orbit o.

For example, the probability of the first successful scan by satellite s_1 in orbit o_1 of task t can be defined as

$$P^{+}_{(t,1,s_1,o_1)} = a_{(s_1,o_1,t)} P_{(s_1,o_1,t)} P^{-}_{(t,0,s_1,o_1)} \tag{2.4}$$

$P^{-}_{(t,0,s_1,o_1)}$ is the probability of failure of all attempted scans before $T^{st}_{(s_1,o_1,t)}$.

If $N^{prev}_{(s_1,o_1)}$ scans were attempted before the current scan, then $P^{-}_{(t,0,s_1,o_1)}$ can be defined as

$$\begin{gathered} P^{-}_{(t,0,s_1,o_1)} = \prod_{N^{prev}_{(s_1,o_1)}} \left[1 - a_{(s^*,o^*,t)} P_{(s^*,o^*,t)}\right] \\ \forall (s^*, o^* \text{ with } T^{st}_{(s^*,o^*,t)} < T^{st}_{(s_1,o_1,t)}) \end{gathered} \tag{2.5}$$

$N^{prev}_{(s_1,o_1)}$ can be calculated using the decision variable $a_{(s,o,t)}$ and the start time corresponding start time ($T^{st}_{(s,o,t)}$).

For the probability of second successful scan by satellite s_1 in orbit o_1 (say) of task t, can be defined as:

$$P^{+}_{(t,2,s_1,o_1)} = a_{(s_1,o_1,t)} P_{(s_1,o_1,t)} P^{-}_{(t,1,s_1,o_1)} \tag{2.6}$$

where, the probability of having only one successful scan before the scan by satellite s_1 in orbit o_1 of task t from the start of the mission horizon, $P^-_{(t,1,s_1,o_1)}$, is defined as:

$$P^-_{(t,1,s_1,o_1)} = \sum_{N^{prev}_{(s_1,o_1)}} \left[a_{(s^*,o^*,t)} P_{(s^*,o^*,t)} P^{(s^*,o^*)-}_{(t,0,s_1,o_1)} \right] \tag{2.7}$$

where, $P^{(s^*,o^*)-}_{(t,0,s_1,o_1)}$ is the probability of failure of all attempted scans before $T^{st}_{(s_1,o_1,t)}$ except the scans by satellite s^* in orbit o^*. Thus the probability of having $d-1$ successful scan before the scan by satellite s_1 in orbit o_1 of task t is defined as:

$$P^-_{(t,(d-1),s_1,o_1)} = \sum_{N^{prev}_{(s_1,o_1)}} \left[a_{(s^*,o^*,t)} P_{(s^*,o^*,t)} P^{(s^*,o^*)-}_{(t,(d-2),s_1,o_1)} \right] \tag{2.8}$$

If the external factors are not taken into consideration, i.e., $P_{(s,o,t)} = 1$ for all (s,o,t), then the probability of d^{th} successful scan depends only on the decision variable $a_{(s,o,t)}$ and the possible values are $\{0,1\}$.

2.3.2 Resource constraints

Satellites have limited resources on-board. The resources include energy and memory specifications of the satellites. These act as the constraints to the optimization problem. The constraints are defined as follows:

2.3.2.1 Energy constraints

Satellites have a finite amount of energy and recharge using solar panels. It is important to ensure that the satellite will have enough energy to complete an operation. It is assumed that the satellite must have $\kappa\,(\geq 0)$ energy remaining after the completion of each operation.

The available charge at the starting time for a scan of task t by satellite s is denoted as $C_{(s,\gamma)}$ where $\gamma = T^{st}_{(s,o,t)}$. The energy $e_{(s,o,t)}$ needed to scan task t in orbit o should always be lesser than or equal than the available charge at the starting time $C_{(s,\gamma)}$ of the scan and the gained energy $ch_s \times \tau_{(s,o,t)}$ during the scan. Therefore, the energy constraint for scanning a task can be written as,

$$\begin{aligned} a_{(s,o,t)}\Big[C_{(s,\gamma)} + ch_s \times \tau_{(s,o,t)} - e_{(s,o,t)}\Big] \geq \kappa \\ \forall t, s, o \text{ when } a_{(s,o,t)} = 1 \end{aligned} \tag{2.9}$$

Another energy constraint is needed to deal with the resources required for the rotation of the satellite. When task $\hat{t}$ is followed by task t, the satellite needs energy to rotate along the roll and pitch angles. It is important to ensure that the satellite has enough energy to rotate. The total available energy at the starting of the rotation is $C_{(s,\gamma)}$ where $\gamma = T^{rot}_{(s,o,t,\hat{t})}$. The energy gained by the satellite during the roll and the pitch is $ch_s\Big(\tau^{roll}_s \theta^{t,\hat{t}}_{(s,o)}\Big)$ and $ch_s\Big(\tau^{pitch}_s \phi^{t,\hat{t}}_{(s,o)}\Big)$ respectively, where $\theta^{t,\hat{t}}$ is the roll angle change needed and $\phi^{t,\hat{t}}$ is the pitch angle change needed while transitioning from task t to task $\hat{t}$. Equation (2.10) describes the energy requirement constraint for transition of the satellite from one task to another.

$$\begin{aligned} & C_{(s,\gamma)} + ch_s\Big(\tau^{roll}_s \theta^{t,\hat{t}}_{(s,o)} + \tau^{pitch}_s \phi^{t,\hat{t}}_{(s,o)}\Big) - \\ & \Big(\theta^{t,\hat{t}}_{(s,o)} e^{roll}(s) + \phi^{t,\hat{t}}_{(s,o)} e^{pitch}(s)\Big) \geq \kappa \quad \forall t \in [1, N^t], \hat{t} \neq t, s, o \end{aligned} \tag{2.10}$$

The last energy constraint is needed to handle the energy requirement for the communication between the satellite and the ground stations. The energy available at $\gamma = T^{sg}_{(s,o,g)}$ is $C_{(s,\gamma)}$. The energy gained during the data transfer is $ch_s \tau^{mem}_s$. The required energy for the communication is $(e^{dl}_s + e^{ul}_s)$. So the constraint for uploading and downloading the data is defined as follows:

$$C_{(s,\gamma)} + ch_s\tau_s^{mem} - (e_s^{dl} + e_s^{ul}) \geq \kappa \qquad \forall g \tag{2.11}$$

e_s^{dl} and e_s^{ul} are considered as the average energy required by a satellite to downlink and uplink the data respectively to the ground station.

2.3.2.2 Memory constraints

Satellites have a finite resource of memory. While scanning, the memory of the satellite fills up and during the communication with the ground station it resets the memory. The satellite s can only perform specific number of scans in between the communication to ground station g and the next available ground station $\hat{g}$. So the memory constrain is defined as,

$$\sum_{N_{(s,g,\hat{g})}} m_{(s,t)} \leq m_s^{max} \qquad \forall g, \hat{g} \neq g \tag{2.12}$$

The value of $N_{(s,g,\hat{g})}$ is calculated as the sum of the decision variable $a_(s,o,t)$ in the interval two communications with the ground station.

2.3.2.3 Constraints related to the start-time of an operation

If the data transfer between the ground station and the satellite is before the scan of task t, the start-time of the scan should be after it has completed the data transfer. So, if a download occurs and satellite takes τ_s^{mem} time to transfer the data, then

$$T_{(s,o,t)}^{st} > \tau_s^{mem} + T_{(s,o,g)}^{sg} \tag{2.13}$$

Similarly, if the data transfer is done after the scan of task t, the data transfer should start after the scan is complete.

$$T^{sg}_{(s,o,g)} > T^{st}_{(s,o,t)} + \tau_{(s,o,t)} \tag{2.14}$$

It is important to ensure that the scanning of the tasks or data transfer operations occur when the satellites are in the VTW of the task and the ground station, respectively. The transfer should occur inside the VTW of the ground station and the scan for the task should always start within the VTW of the tasks, i.e.,

$$\begin{aligned} T^{VTW_{st}}_{(s,o,g)} \leq T^{st}_{(s,o,g)} \leq T^{VTW_{end}}_{(s,o,g)} \\ T^{VTW_{st}}_{(s,o,t)} \leq T^{st}_{(s,o,t)} \leq T^{VTW_{end}}_{(s,o,t)} \end{aligned} \tag{2.15}$$

While transitioning from one task to another, before starting the second task, the satellite needs to assure that it has completed the scan for the first task and has enough time to rotate. Hence, to ensure the minimum time interval between start-time for scan of task t and start-time for scan of $\hat{t}$, the constraint is defined as

$$\begin{aligned} T^{st}_{(s,o,\hat{t})} \geq & T^{st}_{(s,o,t)} + \tau_{(s,o,t)} + \\ & \tau^{roll}_{s} \theta^{t,\hat{t}}_{(s,o)} + \tau^{pitch}_{s} \phi^{t,\hat{t}}_{(s,o)} \end{aligned} \tag{2.16}$$

Therefore, along with the constraints from (2.9) to (2.16) and the objective function (2.1) the optimization problem is defined as a mixed integer non-linear programming problem

$$\underset{\left(a_{(s,o,t)}, T^{st}_{(s,o,t)}, T^{sg}_{(s,o,g)}\right)}{\arg\max} \sum_{t=1}^{N^t} \sum_{d=1}^{D^{max}_t} R^f_{(t,d)} \rho_{(t,d)} \tag{2.17}$$

Subject to,

$$a_{(s,o,t)}\Big[C_{(s,\gamma)} + ch_s\tau_{(s,o,t)} - e_{(s,t)}\Big] \geq \kappa \;\; \forall t \in [1, N^t] \tag{2.18}$$

$$\begin{gathered} C_{(s,\gamma)} + ch_s\Big(\tau_s^{roll}\theta_{(s,o)}^{t,\hat{t}} + \tau_s^{pitch}\phi_{(s,o)}^{t,\hat{t}}\Big) - \\ \Big(\theta_{(s,o)}^{t,\hat{t}}e^{roll}(s) + \phi_{(s,o)}^{t,\hat{t}}e^{pitch}(s)\Big) \geq \kappa \;\; \forall t \in [1, N^t], \hat{t} \neq t, s, o \end{gathered} \tag{2.19}$$

$$C_{(s,\gamma)} + ch_s\tau_s^{mem} - (e_s^{dl} + e_s^{ul}) \geq \kappa \qquad \forall g \tag{2.20}$$

$$\sum_{N_{(s,g,\hat{g})}} m_{(s,t)} \leq m_s^{max} \qquad \forall g, \hat{g} \neq g \tag{2.21}$$

$$T_{(s,o,t)}^{st} > \tau_s^{mem} + T_{(s,o,g)}^{sg} \tag{2.22}$$

$$T_{(s,o,g)}^{sg} > T_{(s,o,t)}^{st} + \tau_{(s,o,t)} \tag{2.23}$$

$$T_{(s,o,g)}^{VTW} \leq T_{(s,o,g)}^{st} \leq T_{(s,o,g)}^{VTW} + V_{(s,o,g)}^{VTW} \tag{2.24}$$

$$T_{(s,o,t)}^{VTW} \leq T_{(s,o,t)}^{st} \leq T_{(s,o,t)}^{VTW} + V_{(s,o,t)}^{VTW} \tag{2.25}$$

$$\begin{aligned} T_{(s,o,\hat{t})}^{st} \geq & T_{(s,o,t)}^{st} + \tau_{(s,o,t)} + \\ & \tau_s^{roll}\theta_{(s,o)}^{t,\hat{t}} + \tau_s^{pitch}\phi_{(s,o)}^{t,\hat{t}} \end{aligned} \tag{2.26}$$

In the next section, a pair of evolutionary algorithms have been proposed for solving the above problem.

2.4 Proposed Solution Techniques

In this section, two heuristic search techniques have been proposed to solve the non-linear mixed-integer programming problem formulated in section 2.3. The proposed algorithms find the allocation of tasks to the satellites and start-time of each operation in the mission horizon. The algorithm finds the schedule of the satellites, which yields the maximum reward. The solution methodology is based on a combination of local search and genetic algorithm with an elitist selection operator.

2.4.1 Elitist Mixed Coded Genetic Algorithm based Satellite Scheduling (EMCGA-SS)

EMCGA-SS is based on a modified elitist genetic algorithm (Kim and Ellis Jr; 2008) with binary and continuous variables together in the structure of the chromosome. EMCGA-SS considers an initial population of candidate solutions, which are random schedules of all the satellites performing the tasks across the whole mission horizon. Our task is to find the near-optimal schedule, that is near-optimal chromosome from the population. The algorithm uses crossover and mutation of schedules to create offspring, and an elitist selection operator is used to select from a pool of offspring schedules. The chromosomes are sorted based on the reward accumulated for the corresponding schedule. In each generation, the best half of the population is chosen for creating the next generation with the help of mutation, crossover and elitist selection operators. The structure of the EMCGA-SS is described in Algorithm 1. Operations of the algorithm are detailed below:

2.4.1.1 Preprocessing

A set of preprocessing operations are performed to make the algorithm computationally efficient. A task may not be present in the VTW of a satellite in all of its revolutions of the satellite trajectory. Depending on the trajectory of the satellite

Algorithm 1: Elitist Mixed Coded Genetic Algorithm based Satellite Scheduling

```
Result: Optimal schedule for all satellites;
Set population of schedules, $\mathcal{P}_0 = \Phi$;
while population size, $|\mathcal{P}_0| \leq \lambda$ do
    Allocate a task randomly to each satellite in each revolution to
      generate random chromosome $p$;
    $\mathcal{P}_0 = \mathcal{P}_0 \cup p$;
end
$Gen = 0$;
while Stopping criteria not reached do
    Sort population $\mathcal{P}_{Gen}$ by reward accumulated (eq. 2.1);
    Select $\lambda/2$ best chromosomes from population, $\mathcal{P}_{Gen}^{\lambda/2}$;
    for $|\mathcal{P}_{Gen+1}| \leq \lambda$ do
        Select 2 parents $p_1$, $p_2$ randomly from $\mathcal{P}_{Gen}^{\lambda/2}$;
        Obtain children $c_1$, $c_2$ by crossover on $p_1$, $p_2$;
        Mutate $c_1$, $c_2$ to obtain $cm_1$, $cm_2$;
        Check feasibility of $c_1, c_2, cm_1, cm_2$ with eq. 2.9-2.16;
        Add best feasible individual to $\mathcal{P}_{Gen+1}$;
    end
    $Gen = Gen + 1$;
end
```

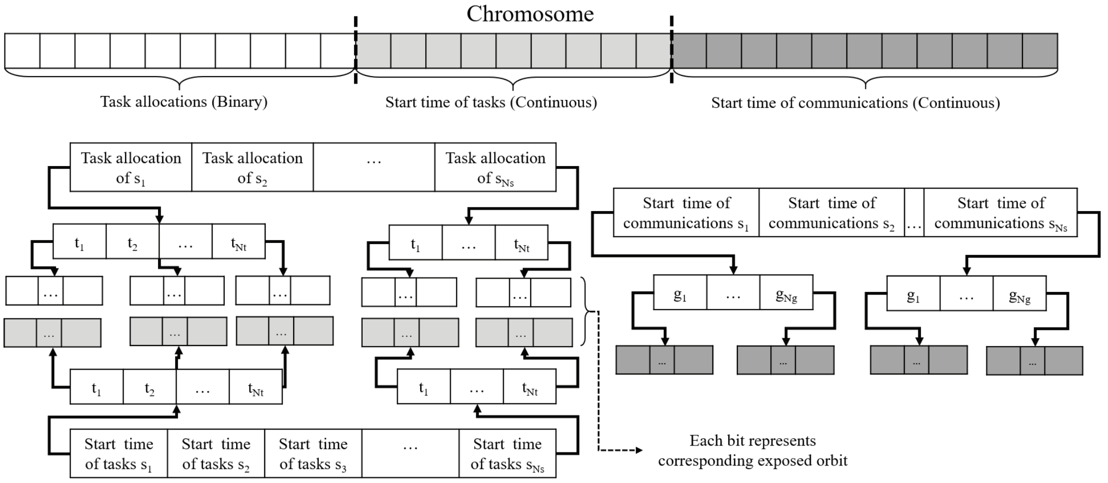

Chromosome
Task allocations (Binary)
Start time of tasks (Continuous)
Start time of communications (Continuous)
Task allocation of s_1
Task allocation of s_2
…
Task allocation of s_{Ns}
t_1
t_2
…
t_{Nt}
Start time of communications s_1
Start time of communications s_2
Start time of communications s_{Ns}
g_1
g_{Ng}
Start time of tasks s_1
Start time of tasks s_2
Start time of tasks s_3
Start time of tasks s_{Ns}
Each bit represents corresponding exposed orbit

Figure 2.4: Structure of chromosome for the algorithms

and the specifications (maximum roll and pitch angle) of the satellite, the revolutions in which a task is exposed to the satellite are calculated. This preprocessing helps to remove the redundant revolutions from the search, where a task is not exposed to the satellite. This makes the algorithm efficient by reducing the size of the chromosomes.

2.4.1.2 Structure of Chromosome

Each chromosome in the population represents the task assignment and start-time of the operations of the satellites. The respective schedules of N_s satellites are juxtaposed to create the chromosome. Since the exposed revolutions for each satellite corresponding to each task calculated in the preprocessing stage are not equal, the number of bits corresponding to the task allocation of each satellite is different. A schedule of a satellite s is designed as a combination of bits containing binary and continuous numbers. The possibility of occurrence of the scan for a task on an exposed revolution $a_{(s,o,t)}$ is the binary bits stored at the beginning of the chromosome. The length of the binary bits are same as the total number of exposed revolutions for all the tasks. The start-time of each corresponding scan $T^{st}_{(s,o,t)}$ is stored in the continuous bits, following the binary bits. The total length of the bits containing start-time is the same as the length of the binary bits. The bits containing the start-time of the communication to the ground station in each exposure $T^{sg}_{(s,o,g)}$, is placed at the end of the schedule of the satellite s. The structure of the chromosome is illustrated in Figure 2.4.

Algorithm 2: Population initialization of EMCHGA-SS

```
Result: Initial schedule of satellites for Algorithm3 ;
Set population of schedules, $\mathcal{P}_0 = \Phi$;
Allocate a task randomly to each satellite in each revolution to generate
  random chromosome $p$;
while population size, $|\mathcal{P}_0| \leq \lambda$ do
    Mutate chromosome $p$ to obtain $p_m$;
    if $p_m$ feasible (eq. 2.9-2.16) then
        Find reward accumulated by $p$ and $p_m$ as $F_p, F_{p_m}$ with eq.2.1;
        if $F_p \geq F_{p_m}$ then
            $\mathcal{P}_0 = \mathcal{P}_0 \cup p$
        else
            $\mathcal{P}_0 = \mathcal{P}_0 \cup p_m$
            $p \leftarrow p_m$
        end
    end
end
```

2.4.1.3 Initial Population

The initial population is created in a systematic way with random initialization of the chromosomes of population size λ. For the binary bits, the values of 0 and 1 are chosen randomly. For the start-time of the scan bits, if the corresponding bit in the task allocation bit-section is 0, it is also set to 0 in the start-time of task bit-section. Otherwise, a random number within the VTW of the specified task in the exposed revolution of a satellite is chosen for the bit value. For the start-time of the communication between the ground station and the satellite, a random number is selected within the VTW of the ground station in each exposure. Therefore, in this process the values of $a_{(s,o,t)}$, $T^{st}_{(s,o,t)}$ and $T^{sg}_{(s,o,g)}$ are initialized in the population.

2.4.1.4 Mutation of a schedule

In order to maintain the genetic diversity, mutation operation is performed on the chromosomes to move from one generation to the next. Here, the mostly used mutation rate $1/m$ is considered (Doerr et al.; 2017) where m is the string size of the chromosome. To mutate, on the binary bit, a random bit from the binary bit section is picked, and the bit value is flipped. For the continuous bits representing the start-time of the scan, the bit corresponding to the bit selected in the binary bit is mutated. If the binary bit is flipped from 1 to 0, the corresponding continuous bit is also set to 0. If the binary bit is set to 1 from 0, a random number from the VTW of the task is chosen as the bit value.

2.4.1.5 Crossover of two schedules

The crossover operation is performed between two mutated offspring. Here, a one-point crossover is performed by selecting a random number. The binary bits and the corresponding continuous bits are interchanged based on the random number generated.

2.4.1.6 Elitist selection operator

To get the best chromosomes in the next generation, an elitist selection operator is used. The selection process is performed based on the reward accumulated by each schedule. The two- parent chromosomes, the mutated children of the parent and the children after the crossover between the mutated children, is compared, and the best chromosome is placed in the next generation.

2.4.1.7 Stopping criteria

The algorithm terminates when the reward accumulated by the best chromosome of the population converges, i.e., it remains unchanged for subsequent generations or a specific time set by the user has elapsed.

In order to deal with the larger chromosome size for the genetic algorithm for a large-scale scenario, a hybrid version of the Hill-climber algorithm and genetic algorithm is proposed. Instead of a randomly generated initial population, an improved initial population is created with Hill-climber as described in the algorithm 2 in order to obtain a faster solution.

2.4.1.8 Hill-climber as initial population

At the first stage, a random chromosome p of the same structure mentioned EMCGA-SS is generated. A mutated chromosome p_m is created and checked if it has a better reward accumulated value than the original chromosome p. If p_m is better then p is discarded and p_m is added to the population of schedule P. Otherwise, the initial random chromosome p is added to P. The process of adding chromosomes is continued until the population size λ is reached. Once an improved initial population is generated with the help of Hill-Climber, this acts as the initial population of the EMCGA-SS. Since the initial population P contains better solutions than randomly generated chromosomes, the optimality is reached faster.

2.5 Simulation

In this section, the problem formulated in section 2.3 is simulated using the proposed solution techniques. Results for both a small-scale scenario and a large-scale scenario are shown. In the formulation, the reward is calculated based on the observation angle of a task from the satellite. A comparison study of the effects of observation angle with respect to accumulated reward is also shown. Scenarios with three, five and eight satellites are also compared to show the performance of multiple satellites in the accumulated reward. A study on the benefits of AEOS over N-AEOS is also shown. Since the formulation is unique to its nature, an exact comparison between other existing frameworks was not possible. However, comparisons between different widely used optimization algorithms like Tabu-Search and Simulated Annealing (Globus et al.; 2004),(Cordeau and Laporte; 2005), (Zhang et al.; 2016), (Wu et al.; 2017), (He, De Weerdt, Yorke-Smith, Liu and Chen; 2018) are shown in this section.

Table 2.1 shows the parameters used in simulation. All the satellites used in the simulation are considered identical.

Table 2.1: Specifications of the satellite considered

Total energy	20
Initial energy	10
Energy for 1deg sensor rotation-roll	0.01
Energy for 1deg sensor rotation-pitch	0.01
Energy required per km of task	0.12
Total memory	50
Memory used per km of task	0.01

2.5.1 Small-scale scenario

In a small-scale scenario, ten tasks are considered. The mission horizon is set to 12 hours. The task parameters are described in Table 2.2. In Table 2.2, the first column represents the task ID, second and third columns represent the corresponding maximum achievable reward and the length of the tasks, respectively. The last column in the table represents the number of scans needed to get the full reward for a particular task (e.g., for task one, three scans are needed to get the full reward. Hence, $R^f_{(1,1)} = 0.5, R^f_{(1,2)} = 0.3, R^f_{(1,3)} = 0.2$).

First, five identical agile satellites with different trajectories have been considered for the initial experiments of the simulation. Later, three and eight satellite scenarios with similar satellite specifications are also included. The satellites have predefined revolutions. From the trajectory of the satellites, it is calculated that one revolution takes approximately 1 hour and 20 minutes for each satellite.

The maximum roll and pitch angle of the satellites are set to 50° and 60°, respectively. By considering these angles, the exposure of each task to a particular satellite on a particular revolution is calculated. A preprocessing is performed to

Table 2.2: Specifications of the considered tasks

ID	Reward	Length (km)	No. of scans required (reward factors)
1	10	100	3 (0.5,0.3,0.2)
2	10	100	4 (0.4,0.3,0.2,0.1)
3	10	120	3 (0.8,0.1,0.1)
4	10	120	3 (0.4,0.4,0.2)
5	25	50	2 (0.7,0.3)
6	10	50	4 (0.4,0.3,0.2,0.1)
7	17	500	4 (0.4,0.3,0.2,0.1)
8	11	400	3 (0.4,0.3,0.3)
9	15	250	3 (0.6,0.3,0.1)
10	20	720	4 (0.4,0.3,0.2,0.1)

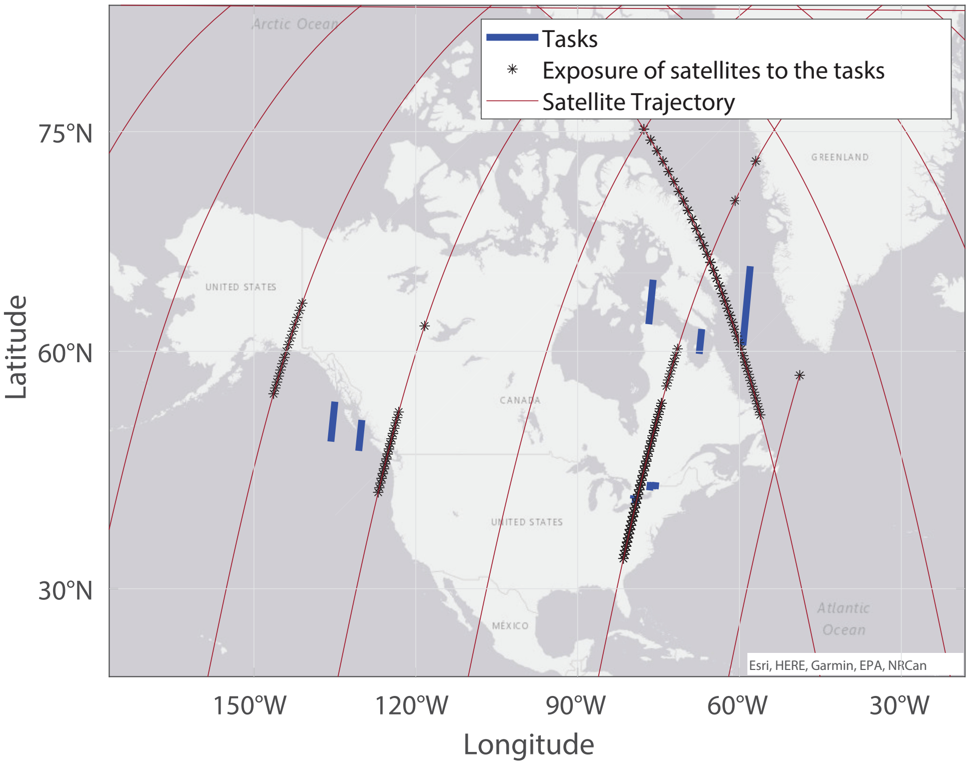


Figure 2.5: Task locations in a small-scale scenario

Table 2.3: Task-wise exposure details of five satellites

Tasks→	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Sat 1	1	1	1	1	1	1	2	3	3	4
Sat 2	1	1	1	1	1	1	1	4	3	4
Sat 3	2	2	2	1	1	1	1	3	4	4
Sat 4	1	1	1	1	1	1	2	3	3	4
Sat 5	1	1	1	1	1	1	2	3	3	4

eliminate redundant revolutions in which no tasks are exposed to the satellite. Table 2.3 illustrates the number of exposures of each task for each satellite. It should be noted that, due to the positional advantages of tasks $8, 9$ and 10 with respect to the satellite trajectories considered, these tasks are exposed to the satellites in a higher number of revolutions than the other tasks. Figure 2.5 shows the task locations for small-scale scenario. It also shows the trajectory of one satellite and its exposures to each task.

In this simulation, the satellites periodically communicate with 6 ground stations. Table 2.4 shows the locations of the considered ground stations in terms of latitude and longitude. While preprocessing, similar to the task exposures, the number of possible communication of the satellites with the ground stations based on the VTW of the ground stations are calculated as well. On the basis of the

Table 2.4: Ground station locations

Name	**Latitude (deg)**	**Longitude (deg)**
Ground station 1	45.58	-75.81
Ground station 2	68.31	-133.5
Ground station 3	53.43	-105.18
Ground station 4	-15.00	30.00
Ground station 5	28.59	77.15
Ground station 6	-7.00	79.00

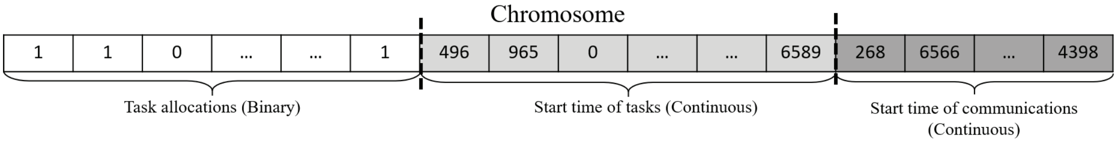

Chromosome
1
1
0
...
...
1
496
965
0
...
...
6589
268
6566
...
4398
Task allocations (Binary)
Start time of tasks (Continuous)
Start time of communications (Continuous)

Figure 2.6: Example structure of a chromosome

exposure of the tasks, the mixed-coded chromosomes for the proposed algorithms are structured. In the entire time horizon, tasks are exposed to satellites for 93 times with considered simulation parameters. Hence, as explained in the Figure 2.4, 93 binary bits are needed for task allocations. For each task allocation bit containing a one, the start-time of tasks are considered from its possible exposures. The start-times are considered as the order of seconds, starting from the beginning of the mission as 0. Here, 93 bits are considered to denote time parameters for the start-time of tasks. The start-time of the communications with the ground station has 125 exposures. Hence, the total number of bits for each chromosome is 93 (binary) $+ 93$ (continuous) $+ 125$ (continuous) $= 311$ for the scenario considered. Figure 2.6 shows a randomly chosen chromosome according to the structure described above. The simulations were performed using *MATLAB 2020a* on a *Intel* $i710^{gen}$ computer with $16GB$ RAM specifications.

Table 2.5: Performance comparison between AEOS and N-AEOS with EMCGA-SS

Satellite Type	**Number of scans**	**Total reward**
AEOS	26	94.46
N-AEOS	21	80.31

A performance comparison of five AEOS and N-AEOS with EMCGA-SS in the considered problem scenario is shown in Table 2.5. For the simulation, all the specifications except the pitch angle for AEOS and N-AEOS are kept same. The

Table 2.6: Total number of scans performed and the rewards accumulated by the satellite with and without considering the observation angle α

	Scans required	**Scans done**	**Max reward**	**Reward with α**	**Reward without α**
T1	3	3	10	7	10
T2	4	3	10	7	9
T3	3	1	10	6	8
T4	3	2	10	3	8
T5	2	2	25	19	25
T6	4	3	10	7	9
T7	4	4	17	13	17
T8	3	3	11	8	11
T9	3	3	15	13	15
T10	4	3	20	13	18

pitch angle is set to zero for the N-AEOS. From the result it is clear that for same task requests, due to higher maneuverability, AEOS can perform higher number of tasks within the mission horizon than N-AEOS and accumulate larger reward. The number of scans performed by five AEOS for all tasks in the mission horizon in a particular Monte-Carlo run is shown in Table 2.6. From the table, it can be seen that some of the tasks like T1, T5, T7, T8 and T9 have been fully completed. But the remaining tasks are incomplete. In our formulation, we have considered that the reward collected by the satellites depends on the observation angle between the task and the satellite. For this simulation, the relationship of the reward with the observation angle is defined by the user. Eq. 4.3 is an example function used for this simulation. Table 2.6 shows that when the observation angle is considered, the full reward is not accumulated by the satellites even when all the scans are complete. However, as per the request of the user and based on the application, if the observation angle does not affect the reward accumulation, the last column in the Table 2.6 shows that the satellites can accumulate the full rewards for those completed tasks.

Table 2.7: Performance comparison for different charging rates

Charging rate	**Scans completed**	**Total reward**	**Engaged time (%)**
0.001	12	68.23	12.20
0.004	14	74.69	12.83
0.009	20	79.54	21.14
0.016	22	84.94	23.24
0.025	23	88.07	23.49
0.036	26	94.46	26.63
0.050	26	94.46	26.63
0.10	26	94.46	26.63

Table 2.7 shows the impact of the energy constraints considered in eq. (2.9-2.11) on the task allocation. When the charging rate increases, the number of scans completed by the satellite increases and the corresponding reward also increases. The number of scans completed and reward get saturated beyond a certain value of the charging rate. The table also shows the amount of time the satellites were engaged in scanning tasks throughout the mission horizon. Even if the satellites have enough energy, due to the overlapping positions of the tasks and lack of task coverage, it is not possible for the satellites to complete all the tasks. The performance of satellites having different charging rates is compared with the *EMCGA-SS* algorithm.

Table 2.8 illustrates the effects of population size for the reward calculation to get the ideal population size which provides the maximize reward. In order to show the benefits of the proposed solution technique, Table 2.9 shows the performance comparison of the proposed solution technique with the widely used Tabu-search and Simulated annealing. *EMCGA-SS* and *EMCHGA-SS* perform better than Tabu-search and Simulated annealing in terms of reward accumulation and the computation time of each runs. On comparing *EMCGA-SS* and *EMCHGA-SS*, *EMCHGA-SS* provides better reward accumulation. Figure 2.7 shows the performance of the algorithms over time with respect to accumulated rewards. It can be seen that *EMCGA-SS* converges faster than other algorithms, whereas Tabu-Search takes the maximum time. From the figure it can be seen that if a very quick solution is needed then *EMCGA-SS* provides a better result than the *EMCGHA-SS*

Table 2.8: Performance comparison for the popuation size for EMCGA-SS

Population Size	**Total reward**
200	84.79
400	87.70
600	91.90
800	94.46
1000	94.51
1200	94.53

The small-scale scenario is again simulated with three satellites and eight satellites. The specifications of the satellites are kept the same, but the trajectories of the satellites are different from each other. Figure 2.8 shows the comparison of the total reward accumulated by three, five and eight satellites. The box plot shows

TABLE 2.9: Performance comparison of different algorithms

Algorithms Considered	Reward	Time needed(Sec)
EMCGA-SS	94.46	68.6424
EMCHGA-SS	**101.36**	96.5565
Tabu Search	89.34	472.8435
Simulated Annealing	86.56	396.5051

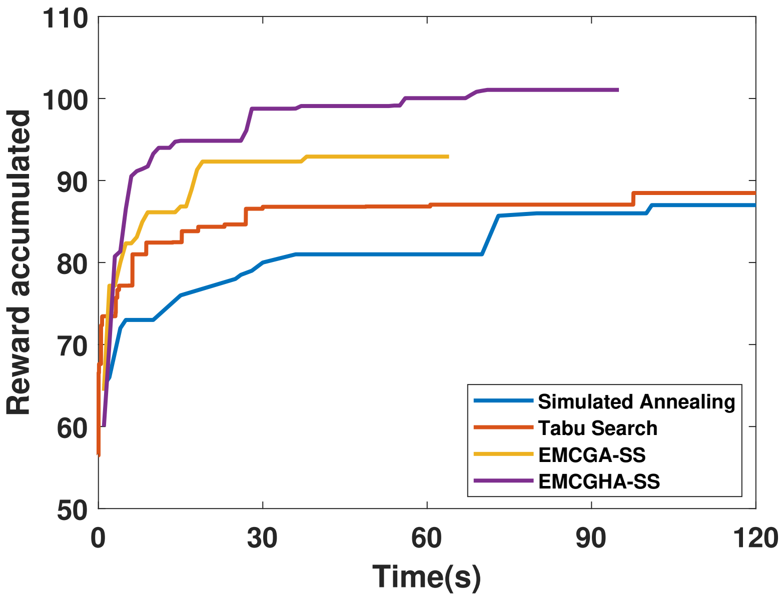


FIGURE 2.7: Performance of the algorithms showing the reward accumulated over time.

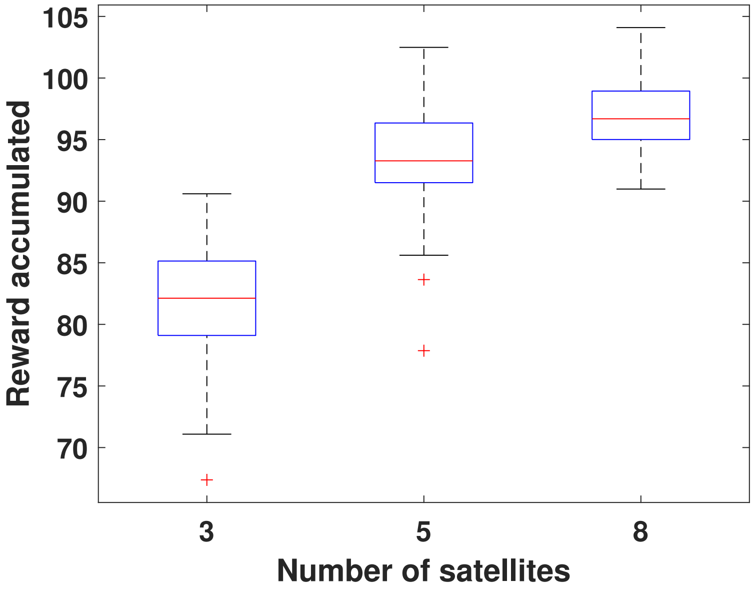


FIGURE 2.8: Reward accumulated with respect to the number of satellites for small-scale scenario

the distribution of reward for 100 Monte-Carlo runs. On each box, the center line represents the median, and the bottom and top edges of the box indicate the 25 and 75 percentiles for the 100 Monte Carlo runs, respectively. The + symbols are the outliers plotted individually.

2.5.2 Large-scale scenario

For a large-scale scenario, a significantly larger task allocation problem with 400 tasks is considered. This section of the simulation provides an insight regarding the optimal task allocation when the number of tasks increases significantly. These tasks have rewards uniformly distributed between 10 and 30. The lengths of these tasks are also uniformly distributed between 30 km and 350 km. For 75% of the considered tasks, it is assumed that one scan is enough to get the full reward, and for the remaining 25% of the tasks, two scans are required to get the full reward with the reward factor of (0.8,0.2).

The considered satellite parameters are kept the same as the small-scale scenario. The structure of the chromosome is modified according to the exposures of the tasks for all of the five satellites. The ground station specifications are also kept the same as the small-scale scenario.

Table 2.10: Performance comparison between AEOS and N-AEOS satellites in large-scale

Satellite Type	**Number of scans**	**Total reward**
AEOS	984	4527
N-AEOS	731	2886

Table 2.11: Performance comparison of different algorithms

Algorithms Considered	**Reward**	**Time needed(Sec)**
EMCGA-SS	4527	1153
EMCHGA-SS	**4867**	1309
Tabu Search	3240	10500
Simulated Annealing	2969	9644

Table 2.10 shows the comparison between the performance of AEOS and N-AEOS using EMCGA-SS. As expected, AEOS performs a higher number of scans during the mission horizon and accumulates larger reward than N-AEOS. For this large-scale scenario, the difference between the performance of AEOS and N-AEOS is clearly more significant than in small-scale scenario.

In Table 2.11, the performance of the proposed algorithms EMCGA-SS and EMCHGA-SS is compared with Tabu-Search and Simulated Annealing for five

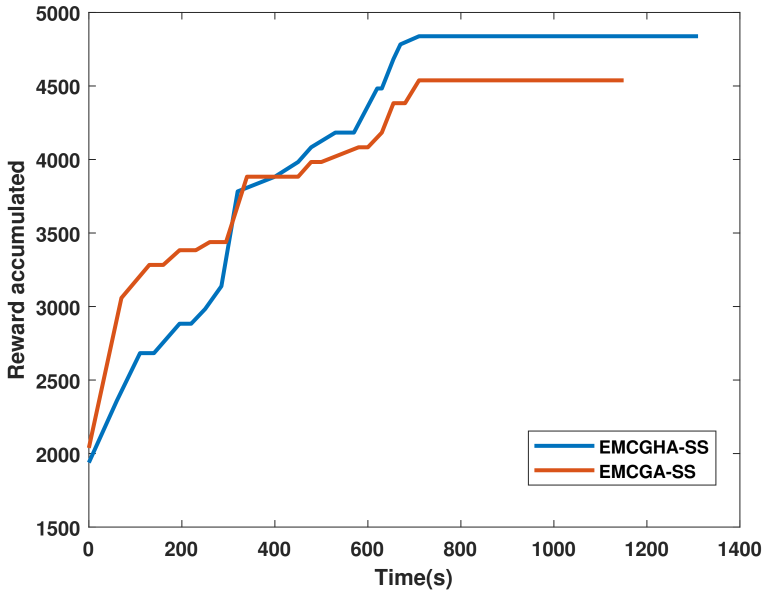


Figure 2.9: Performance comparison of EMCGA-SS and EMCGHA-SS showing the reward accumulated over time.

AEOS satellites. This table for a large-scale scenario shows the same trend as obtained in the small-scale scenario. EMCHGA-SS provides a better schedule in terms of total reward accumulated whereas EMCGA-SS provides the solution with lesser computation time.

Figure 2.9 shows a performance comparison between the proposed algorithms in the large scale scenario. From the graphs it is visible that if a user wants a quicker solution EMCGA-SS provides a better solution than EMCGHA-SS. When time is not an important factor EMCGHA-SS yields solution with more rewards. Comparing 2.7 and 2.9, it can be seen that in the small scale scenario, as the number of bits in a chromosome is smaller the advantage of using only EMCGA-SS is not that significant but in a large scale scenario, if time is a constraint for the user then by using EMCGA-SS over EMCGHA-SS can provide a better result.

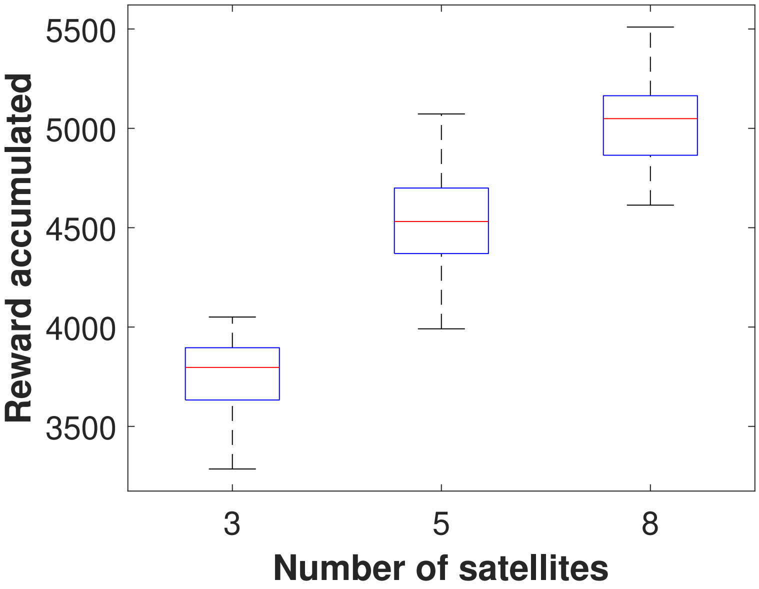


FIGURE 2.10: Reward accumulated with respect to the number of satellites for large-scale scenario

A comparison between three, five and eight satellites is also performed for 100 Monte-Carlo runs and the results are shown in Figure 2.10. The figure with box plots shows that if the number of satellites increases the reward accumulation increases as well in the large-scale scenario.

2.6 Conclusion

A realistic formulation of an AEOS scheduling problem and a sub-optimal solution technique are proposed in this work. To find the optimal schedule of the tasks specified by the user and the start-time of the operations, a reward is associated with the tasks. The proposed mixed-integer non-linear optimization problem finds the optimal schedule by maximizing the reward with the realistic satellite resources as energy and memory constraints. To incorporate the need of multiple scans to complete a task, a reward factor is included in the objective function. Probability-based failure and success rates for completion of scanning the tasks is also taken into consideration in the proposed model. An elitist mixed coded genetic algorithm-based methodology has been developed to solve the proposed scheduling model. This algorithm is further extended with a hill-climber based initialization method. A number of scenarios are considered for testing the efficiency of the algorithms. It has been concluded that the proposed scheduling algorithms produce better results than the most popular techniques like Tabu-search and Simulated annealing in terms of calculated rewards and computational time. It has also been reconfirmed quantitatively through the proposed study that AEOS is a better technology than N-AEOS.

In this work, we consider that a task region can be scanned fully in one visit. For future work, it can be considered that if a task region is too large for the satellites to scan in one go, an optimal segmentation of the task region is required or an optimal schedule algorithm is required to deal with the large region.

Bibliography

Baek, S.-w., Han, S.-m., Cho, K.-r., Lee, D.-w., Yang, J.-s., Bainum, P. M. and Kim, H.-d. (2011). Development of a scheduling algorithm and GUI for autonomous satellite missions, *Acta Astronautica* **68**(7-8): 1396–1402.

Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G. and Raymond, V. (2007). A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites, *European Journal of Operational Research*

177(2): 750–762.

Bianchessi, N. and Righini, G. (2008). Planning and scheduling algorithms for the COSMO-SkyMed constellation, *Aerospace Science and Technology* **12**(7): 535–544.

Chu, X., Chen, Y. and Tan, Y. (2017). An anytime branch and bound algorithm for agile earth observation satellite on-board scheduling, *Advances in Space Research* **60**(9): 2077–2090.

Cordeau, J.-F. and Laporte, G. (2005). Maximizing the value of an earth observation satellite orbit, *Journal of the Operational Research Society* **56**(8): 962–968.

Dilkina, B. and Havens, B. (2005). Agile satellite scheduling via permutation search with constraint propagation, *Technical Report Actenum Corporation* .

Doerr, B., Le, H. P., Makhmara, R. and Nguyen, T. D. (2017). Fast genetic algorithms, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 777–784.

Globus, A., Crawford, J., Lohn, J. and Pryor, A. (2004). A comparison of techniques for scheduling earth-observing satellites, *Applications of Artificial Intelligence Conference*.

Habet, D., Vasquez, M. and Vimont, Y. (2010). Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite, *Computational optimization and applications* **47**: 307–333.

He, L., De Weerdt, M., Yorke-Smith, N., Liu, X. and Chen, Y. (2018). Tabu-based large neighbourhood search for time-dependent multi-orbit agile satellite scheduling, *Proc. of the ICAPS* **18**: 45–52.

He, L., Liu, X., Laporte, G., Chen, Y. and Chen, Y. (2018). An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling, *Computers & Operations Research* **100**: 12–25.

Ju, J. and Roy, D. P. (2008). The availability of cloud-free landsat ETM+ data over the conterminous united states and globally, *Remote Sensing of Environment* **112**(3): 1196–1211.

Kim, J.-L. and Ellis Jr, R. D. (2008). Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling, *Journal of construction engineering and management* **134**(11): 904–913.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M. and Bataille, N. (2000). How to manage the new generation of agile earth observation satellites, *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Citeseer, pp. 1–10.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M. and Bataille, N. (2002). Selecting and scheduling observations of agile satellites, *Aerospace Science and Technology* **6**(5): 367–381.

Liao, D.-Y. and Yang, Y.-T. (2007). Imaging order scheduling of an earth observation satellite, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(5): 794–802.

Lin, C.-H., Tsai, P.-H., Lai, K.-H. and Chen, J.-Y. (2012). Cloud removal from multitemporal satellite images using information cloning, *IEEE transactions on geoscience and remote sensing* **51**(1): 232–241.

Liu, X., Laporte, G., Chen, Y. and He, R. (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time, *Computers & Operations Research* **86**: 41–53.

Lu, J., Chen, Y. and He, R. (2020). A learning-based approach for agile satellite on-board scheduling, *IEEE Access* **8**: 16941–16952.

Pang, C. K., Kumar, A., Goh, C. H. and Le, C. V. (2015). Nano-satellite swarm for SAR applications: Design and robust scheduling, *IEEE Transactions on Aerospace and Electronic Systems* **51**(2): 853–865.

Peng, G., Song, G., Xing, L., Gunawan, A. and Vansteenwegen, P. (2020). An exact algorithm for agile earth observation satellite scheduling with time-dependent profits, *Computers & Operations Research* p. 104946.

Tangpattanakul, P., Jozefowiez, N. and Lopez, P. (2012). Multi-objective optimization for selecting and scheduling observations by agile earth observing satellites, *International Conference on Parallel Problem Solving from Nature*, Springer, pp. 112–121.

Tharmarasa, R., Chatterjee, A., Wang, Y., Kirubarajan, T., Berger, J. and Florea, M. C. (2019). Closed-loop multi-satellite scheduling based on hierarchical MDP, *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–7.

Tharmarasa, R., Kirubarajan, T., Berger, J. and Florea, M. C. (2019). Mixed open-and-closed loop satellite task planning, *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–8.

Wang, X., Song, G., Leus, R. and Han, C. (2019). Robust earth observation satellite scheduling with uncertainty of cloud coverage, *IEEE Transactions on Aerospace and Electronic Systems* **56**(3): 2450–2461.

Wang, X.-W., Chen, Z. and Han, C. (2016). Scheduling for single agile satellite, redundant targets problem using complex networks theory, *Chaos, Solitons & Fractals* **83**: 125–132.

Wolfe, W. J. and Sorensen, S. E. (2000). Three scheduling algorithms applied to the earth observing systems domain, *Management Science* **46**(1): 148–166.

Wu, G., Wang, H., Pedrycz, W., Li, H. and Wang, L. (2017). Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy, *Computers & Industrial Engineering* **113**: 576–588.

Zhai, X., Niu, X., Tang, H., Wu, L. and Shen, Y. (2015). Robust satellite scheduling approach for dynamic emergency tasks, *Mathematical Problems in Engineering* **2015**.

Zhang, Y., Wu, J., Cai, Z., Zhang, P. and Chen, L. (2016). Memetic extreme learning machine, *Pattern Recognition* **58**: 135–148.

Chapter 3

Multi-Stage Optimization Framework of Satellite Scheduling for Large Areas of Interest

The content of this chapter is a first revision of the manuscript text for publication in Advances in Space Research under the following citation:

Chatterjee, Abhijit and Tharmarasa, Ratnasingham (2023) Multi-stage optimization framework of satellite scheduling for large areas of interest, *First revision under review, Advances in Space Research, Elsevier.* (First revision submitted in August 2023)

Multi-Stage Optimization Framework of Satellite Scheduling for Large Areas of Interest

Abstract

The satellites are assigned to perform scanning tasks by maneuvering sensors in roll and pitch directions within resource constraints. However, these task regions may be larger than the scanned region by the satellites in a single attempt, so multiple attempts are needed to complete the task. Previous literature has proposed strip-based segregation for larger task regions with parallel strips. However, satellite trajectories may not be parallel to each other, and multiple visits from multiple satellites may be required to complete a scanning task. This results in higher usage of resources due to overlapping strips in strip-based segregation. Therefore, a novel three-stage scheduling methodology is proposed for better performance in terms of the size of the scanned region and resource utilization. The initial stage assigns satellites to the tasks irrespective of task segmentation for each scan. The second stage optimizes the roll and pitch angles of assigned satellites to maximize the scanned region with parallel computation since the tasks are independent. The third stage handles uncertainty associated with task failures. An elitist mixed-coded evolutionary solution technique with a constrained non-linear optimization and Markov decision process is proposed. The efficiency of the proposed methodology in small and large-scale scenarios is illustrated with simulations.

Keywords: *Satellite scheduling, Large region scheduling, Multi-stage optimization, Markov decision process, Genetic Algorithm*

3.1 Introduction

Agile Earth Observing Satellites (AEOS) scan the Earth from space to perform specific tasks such as surveillance, tracking, and monitoring natural disasters and environmental changes using the various onboard sensors. AEOSs perform a series of scanning tasks based on the requests from the users. These task locations might be spread across large regions in a scattered manner on the Earth's surface.

The location of the AEOS determines the visible window (VW) of the AEOS based on sensor specifications (maximum roll and pitch angles). The task locations can only be scanned when they belong to the VW of the AEOS. When a request to scan a large region is received, the request either involves the region to be scanned in a geometrically consistent manner or involves scanning the whole region within a specified time window. This paper is focused on the latter scenario. An optimal assignment of a sequence of tasks to respective AEOSs by satisfying the resource constraints is necessary to complete the user requests in the specified time frame (Hall and Magazine; 1994).

The task Areas of Interest (AOIs) for the AEOSs are usually classified as spot and polygon targets. Spot targets or polygon targets with smaller AOIs have a limited area to scan, which can be completed by one pass according to the observation scope of the AEOS sensor, while larger polygon targets may need to be captured by multiple passes of multiple AEOSs in a cooperative manner. A large portion of the literature focuses on spot targets, comparatively smaller polygon targets or sub-tasks with smaller regions originating from large polygon targets (Habet et al.; 2010; Wang et al.; 2020). AEOS scheduling for spot target-based scanning tasks is relatively easier as the length or the orientation of the task length does not need to be considered in the problem (Renjie et al.; 2008). Based on the task requests from the users, larger AOI scanning is an extremely relevant and realistic problem to address, as in applications like monitoring forest fires, crop growth for large regions, or iceberg melting rates. When the AOIs are too large for AEOSs to scan in one go, the complexity of scheduling multiple AEOSs in their multiple revolutions to perform multiple scans for each task increases manifold (Niu et al.; 2018).

The swath width of a sensor in the AEOS determines the maximum area that can be scanned in one pass. Since the specifications of the AEOS, such as the swath width can not be altered, an acceptable solution to fulfill the requirements of scanning a large region is to divide it into some smaller regions which can be scanned in a single pass. In order to achieve maximum efficiency, an optimal division of the large region is necessary. A vast majority of studies have considered a strip-wise division of a large region, where the strips are parallel to the trajectory of the AEOS (Niu et al.; 2018; Cordeau and Laporte; 2005; Xu et al.; 2018, 2020). However, the trajectories of the AEOSs are predefined, usually independent, and not parallel to each other. Hence, dividing the strips parallel to the satellite trajectories results in the overlapping of the strip regions. When the AEOS performs scanning operations on those strips according to the schedule, they will need to utilize more AEOS resources (energy, memory) by scanning the overlapped region multiple times (Gu et al.; 2022).

In order to minimize the overlapping of pre-defined strips, a method of further division of the strips into disjoint areas has been illustrated, which results in a significant amount of overlaps nevertheless (Berger; 2016). In the present work, the overlaps are minimized further by not considering pre-defined strips for the large area division. Instead, the proposed method finds the optimal roll and pitch angle for each of the multiple assigned scans before scanning and accordingly creates the segments of smaller regions by minimizing the overlapped regions. This method helps in quantifying the overlapped region. The proposed method is illustrated in the following sections to have no overlaps but higher coverage. The overlapped regions in moderate quantities while scanning could be beneficial in commercial systems like Attitude Determination and Control Systems (ADCS) to account for image stitching for a large region and compensate for missing regions at the seams from each scan of the smaller regions. A possible future scope of the current work will be considering an application-oriented fixed quantity of necessary overlapping in scans to account for compensation errors.

For scanning the task, the observation angle (which is the combination of roll and pitch angles) of the AEOS corresponding to the task needs to be determined. The advantage of both roll and pitch angles provides AEOS a larger Visible Time Window (VTW) for performing tasks compared to Non-AEOS (N-AEOS). In this work, the determination of an optimal observation angle is considered for each task.

After completing the scan for each small region successfully, the AEOS receives a reward defined by the user. The user can also specify the portion of the reward associated with the percentage of the area scanned for the whole AOI. In literature, a reward-based optimization problem is often formulated, where maximizing the reward provides an optimal task schedule for the AEOSs (Han et al.; 2022). In (Peng et al.; 2020), a reward function depending on the start time, which affects the observation angle of the scan, is considered. A reward function based on the quality of the image is used in (Wolfe and Sorensen; 2000). A ten-level scale for image quality-based reward for each task has been introduced in (Liu et al.; 2017) as a constraint for maximizing the total reward obtained for scanning the tasks.

The quality of a scan might depend on the observation angle as well. A reward function based on the observation angle is considered in the problem formulation of this present work. Depending on the nature of a task, multiple scans for the same area might be needed in order to complete that task and receive a full reward. A step-function-based reward factor is used to deal with the multiple scans requirement for a task for N-AEOS scheduling in (Tharmarasa, Kirubarajan, Berger and Florea; 2019) and for AEOS problems (Chatterjee and Tharmarasa; 2022). Due to additional challenges, in this paper, it is assumed that multiple scans of the same region inside the task AOI are not needed to get the whole reward.

The AEOS receives the reward for completing a scan only if the scan is successful. There are many contributing factors like cloud coverage and sensor malfunctions during missions resulting in a scan failure (Wang et al.; 2019). In applications like surveillance or tracking, cloud coverage hinders the performance of the total mission considerably. A real-life data set of *Landsat-7* showed 35% of the captured images were covered with cloud (Ju and Roy; 2008). In (Lin et al.; 2012), an image processing-based cloud removal technique is proposed where the main focus was put into post-processing of the scanned region. A semi-Markov Decision Process (MDP) formulation has been introduced in (Eddy and Kochenderfer; 2020), where the maneuverability of the satellites has not been considered. In (Zhai et al.; 2015), it was proposed that no reward would be allocated when there is uncertainty regarding the performance of the AEOS, while some researchers proposed a probability-based success and failure for the scans of the AEOS (Han et al.; 2022). In the open-loop approach of satellite scheduling, a predicted uncertainty factor (often considered to be stochastic arising due to conditions like cloud coverage or sensor malfunctions) is assumed before the start of the mission, and the scheduling is done pre-mission. In the closed-loop approach, the information obtained during the mission is used to further optimize the consecutive scans. An MDP-based reinforcement learning approach is shown in (Wen et al.; 2023), where only one satellite is considered for the schedule. The MDP-based approach in (Tharmarasa, Chatterjee, Wang, Kirubarajan, Berger and Florea; 2019) illustrates uncertainty handling in a closed-loop framework, and concludes that a high-dimensional state space of the order of $10E5$ is needed even for a problem with 7 tasks and 3 satellites. In this work, another MDP-based approach is introduced, where the pre-mission information is used to fine-tune the roll and pitch angles for each assigned scan. As the tasks are considered independent of each other, multiple smaller MDP problems are used in parallel to reduce the number of states drastically. The information of a failure during a mission from an AEOS is used to find an updated greater coverage. With the contribution of all the factors mentioned above, an optimization model is developed to find the optimal schedule for the AEOSs.

The choice of an appropriate optimization algorithm depends on the chosen parameters and the optimization constraints. Several methods of permutation-based space search to tackle the complexity of an AEOS scheduling problem, including simulated annealing, hill-climber, and genetic algorithms with various mutation operators are studied in (Globus et al.; 2004). In (Lemaître et al.; 2002), the authors showed the comparison between a greedy algorithm, local search, a dynamic programming algorithm and a constraint programming approach for AEOS scheduling problem. A metaheuristic approach based on adaptive large neighborhood search is illustrated to perform time-dependent transition time consideration for AEOS scheduling (Liu et al.; 2017). While this work was proposed for single AEOS scheduling, in (He et al.; 2018), it was extended to multiple AEOS scheduling.

A complex network theory-based approach has been implemented in an AEOS scheduling problem (Wang et al.; 2016). However, the complexity of multiple AEOS scheduling becomes near impossible to solve with network theory. A genetic algorithm-based solution methodology has been proposed in (Wu et al.; 2022) for a single AEOS scheduling problem. When the problem is extended to a multi-AEOS problem, the complexity of the chromosomes increases. An elitist-mixed coded genetic algorithm (EMCGA)-based approach is used in (Chatterjee and Tharmarasa; 2022) for multi-AEOS problems with smaller AOI targets. In this paper, a modified approach to the EMCGA algorithm has been introduced since the resource constraints are similar in nature.

This work proposes a multi-stage optimization framework for scheduling large imaging tasks with multiple AEOSs prior to the start of the mission. The important factors considered in this optimization model are the assignment of the multiple AEOSs to each user-specified task, the consideration of the start time and duration of the scans to ensure completion in the required time frame, and the choice of roll and pitch angles of the AEOS for each scan. A detailed overall optimization problem is discussed at first, then that problem is segmented into a multi-stage optimization problem where these factors are handled in three stages to reduce the complexity of each stage of the optimization problem. The first stage of the optimization problem handles the assignment of the tasks, start time, and duration of the scan, while the second stage finds the optimal roll and pitch angle for each assignment. The third stage is used to handle the uncertainty of the success of a scan.

In the following section 3.2, the problem statement is discussed. The mathematical formulation of the problem is discussed in section 3.3. Sections 3.4 and 3.5 illustrate the solution methodology for this multi-stage optimization framework and experimental analyses with simulations using small and large-scale scenarios. In section 3.6, the conclusions of the work and possible future directions are discussed.

3.2 Problem Statement

The main focus of this paper is to identify an optimal allocation of multiple AEOSs in the mission to perform the observation *tasks* within a particular user-specified time frame by considering AEOS resource constraints and task specifications.

These user-specified tasks involve imaging specific geographical locations. The task regions can be of limited geographical dimensions (defined as spot targets) or widespread polygonal regions (defined as massive targets). In this work, it is assumed that the task AOIs can be significantly larger than the visibility range of any AEOS while passing over it. These large tasks would need multiple *scans* of smaller regions that could be completed by a single AEOS visit. These user-requested tasks are known prior to the start of the mission.

To ensure task completion, rewards are assigned in the optimization process, depending upon the task success percentage. In order to complete imaging the whole task AOI successfully and achieve the full reward, multiple visits by the same or different AEOSs are needed. An optimal schedule of the considered multiple AEOSs needs to be determined to complete scanning each of the AOIs to the fullest possible extent corresponding to the user requests. This scheduling process is performed prior to the start of the mission when the requested tasks are known.

The reward for a scan is calculated by the amount of area covered by an AEOS in a single visit. As multiple scans are needed to cover the whole region of the task, the total reward accumulated for each task is the summation of the reward accumulated during each individual scan. The reward for each scan is assigned when the AEOS completes a scan successfully.

Based on the task specifications provided by the user, there can be a requirement to complete all the scans for a task within a particular time frame. For example, in timed surveillance tasks, the user provides a time window to complete the whole task. The completion of the tasks needs to be within the specified time window.

To ensure that the timing of the scans is appropriate, penalties are assigned when the task's time-frame requirements are not satisfied. Since multiple visits by the same or different AEOS might be needed for a task, the AEOS assignment takes into consideration the overlap of the visible time windows of the task for different visits and the user-requested time frame for task completion. The AEOS performs each scan by adjusting the roll and pitch angles of the sensors while maintaining the heading in its trajectory. Figure 2.3 shows the roll and pitch angles w.r.t to a task and AEOS. The optimal roll and pitch angles corresponding to each AEOS assigned to a particular AOI need to be determined on every scan to ensure the maximum coverage of the AOI. The swath width or the visibility of the satellite is considered to be small enough so that the earth's curvature can be ignored.

The AEOSs are considered to have limited memory capacity. Hence, periodically, the AEOSs need to transfer the collected data to the ground stations to free up memory space. While communicating with the ground stations, the satellites downlink all the memory collected to the ground stations. The AEOS can not perform any scan during communication with the ground stations.

3.3 Problem formulation

The objective of the problem is to maximize the reward accumulated by the N_s AEOSs by performing scanning tasks on user-specified regions. The task regions are larger than the coverage of the scanning capacity of an AEOS in one attempt. The proposed formulation is also valid if some of the task regions are smaller than the scanning capacity of an AEOS. The variables for the optimization problem are defined as follows,

- $x(s, o, t) \in \{0, 1\}$ – decision variable denoting the assignment of AEOS s to task t in revolution o. Here 0 means that there are no assignments and 1 means at least 1 assignment.

- $T_{st}(s, o, t)$ – a vector decision variable containing start time for all the assigned scans of AEOS s in revolution o to task t. If there are continuous scans of task t with different roll and pitch angles, those are considered discrete elements, and the start times are noted.

- $t_{st}(s, o, t, i)$ – decision variable denoting ith element for $T_{st}(s, o, t)$.

- $\Theta(s, o, t)$ – vector decision variable denoting the roll angle for the scan of task t by AEOS s during revolution o. The size of the vector would be the same as $T_{st}(s, o, t)$.

- $\Phi(s, o, t)$ – vector decision variable denoting the pitch angle for the scan of task t by AEOS s during revolution o. The size of the vector would be the same as $T_{st}(s, o, t)$.

- θ_t^i – decision variable denoting roll angle for ith scan of task t.

- ϕ_t^i – decision variable denoting pitch angle for ith scan of task t.

- $D(s, o, t)$ – a vector decision variable containing duration for all the assigned scans of AEOS s in revolution o to task t. The size of the vector would be the same as $T_{st}(s, o, t)$.

- $d(s, o, t, i)$ – decision variable denoting ith element for $D(s, o, t)$, which is the duration of the ith scan for task t. It can take any value from 0 to the maximum duration required for a scan using satellite s.

- $T_{(k,t)}^{start}$ – start time of the kth scan for task t including all the AEOSs and all their revolution calculated from $T_{st}(s, o, t)$.
- d_t^k – duration of scan for kth scan of task t.
- A_t – total area of task t.
- $\tilde{A}(s, o, t, i)$ – area of task t scanned by AEOS s in revolution o in the ith scan.
- $\alpha(t)$ – earliness penalty cost for task t per unit time step.
- $\beta(t)$ – tardiness penalty cost for task t per unit time step.
- $R_{max}(t)$ – the maximum reward for the task t provided by the user.
- $\mathbb{R}_{unit}^t$ – reward for scanning unit area of task t.
- $\tilde{R}(s, o, t, i)$ – the reward obtained for successful completion of ith scan by AEOS s for task t in revolution o.
- $\mathbb{R}_{unit}(s, o, t, i)$ – unit reward for ith scan by AEOS s in revolution o for task t including the penalty.
- $\zeta(t, \theta, \phi)$ – the reward factor for the AEOSs' roll and pitch angles during the scan.
- N_t – the number of tasks specified by the users

- N_s – the number of AEOSs in the mission
- N_{os} – number of revolutions performed in the trajectory of AEOS s in the given time horizon.
- N_i^t – the number of assigned scans for task t.
- $N_{s,g,\hat{g}}$ – number of scans by AEOS s between communication with ground station g and $\hat{g}$
- θ_t^i – decision variable denoting roll angle for ith scans of task t.
- ϕ_t^i – decision variable denoting pitch angle for ith scans of task t.
- Θ_{max}, Φ_{max} – maximum roll and pitch angles for AEOSs
- ψ – swath width angle of AEOSs
- $m(s, o, t, i)$ – memory used for ith scan of task t by AEOS s in revolution o.
- m_{unit} – memory used to scan per unit time step.
- m_{max} – maximum memory capacity of an AEOS.
- $T_g(s, o)$ – communication start time to ground station g for satellite s in revolution o.
- $d_g(s, o)$ – communication duration to ground station g for satellite s in revolution o.
- $\tilde{B}^t$ – Set of overlapped region for task t due to multiple scans.

- $\mathbb{B}_n^t$ – Reward accumulated for the nth overlapped scan for task t.

In the following equations, maximize has been used to define the maximization of an objective function whereas, max has been used to define the maximum of two components. When the AEOSs perform scanning tasks, some scans need to be initiated within a specified time window defined by the user. If the scanning is done before or after the specified time window, an earliness or a tardiness penalty is applied, respectively. Let the scanning of a task t have a requirement to be started within a particular time window, defined as $\left[\tau_t^l,\ \tau_t^u\right]$ where τ_t^l and τ_t^u denote the start time and the end time of the time window, respectively. As discussed in section 3.2, multiple scans might be needed to cover the whole region for the task t. If a scan of task t starts earlier than τ_t^l, then an *earliness penalty* is incurred and, if it starts later than τ_t^u, then a *tardiness penalty* is incurred. Let $t_{st}(s,o,t,i)$ denote the start time of the i^{th} scan of task t by AEOS s in revolution o. Then the earliness $\mathcal{E}(s,o,t,i)$ and tardiness $\mathcal{L}(s,o,t,i)$ factors are calculated as follows,

$$\mathcal{E}(s,o,t,i) = \max\left(\tau_t^l - t_{st}(s,o,t,i), 0\right) \tag{3.1}$$

$$\mathcal{L}(s,o,t,i) = \max\left(t_{st}(s,o,t,i) - \tau_t^u, 0\right) \tag{3.2}$$

To fulfill the requirements of task t, all the scans required may not be performed by a single AEOS in a single revolution. The reward achieved by scanning the area $\tilde{A}(s,o,t,i)$ in the ith scan for task t by AEOS s in revolution o will get penalized if it does not occur within the specific time window $\left[\tau_t^l,\ \tau_t^u\right]$. Since this reward accumulated is to be maximized, either of the penalties due to earliness and tardiness incurred would be subtracted from the reward. Here, $\alpha(t)$ and

$\beta(t)$ are the associated earliness and tardiness costs per unit of the time factors, respectively, for task t. This penalty would be effective only on the area scanned by the AEOS $\tilde{A}(s,o,t,i)$. The reward achieved by scanning might depend on the observation angle, which is the combination of roll (θ) and pitch (ϕ) angles for the position of task t from the AEOS s. The relationship of reward with observation angle is specific to the task and defined by the user. In general, the reward is inversely proportional to the observation angle, i.e., when an AEOS performs the scans directly above the task region, with 0° roll and pitch, it gathers the maximum reward (Liu et al.; 2017). The swath width of the satellite is considered to be small enough that the earth's curvature does not affect the area scanned by the AEOS. An analysis is done in (Chatterjee and Tharmarasa; 2022), where the effect of reward corresponding to the change in observation angle is shown. The observation angle is calculated as a combination of the roll and pitch angle movements of the AEOS, depending on the position of the task. A reward factor corresponding to the roll and pitch angles, $\zeta(t,\theta_t^i,\phi_t^i)$, is multiplied in the reward calculation to address this issue. If the reward does not depend on the angle of the scan, $\zeta(t,\theta_t^i,\phi_t^i)$ can be considered as 1. Considering the above factors, the reward accumulated by AEOS s in revolution o for ith scan for task t can be given as,

$$\begin{aligned}\tilde{R}(s,o,t,i) &= \mathbb{R}_{unit}^t \tilde{A}(s,o,t,i)\zeta(t,\theta_t^i,\phi_t^i) - \left[\frac{\tilde{A}(s,o,t,i)}{A_t}\left(\alpha(t)\mathcal{E}(s,o,t,i)\right.\right. \\ &\quad \left.\left. + \beta(t)\mathcal{L}(s,o,t,i)\right)\right] \\ &= \tilde{A}(s,o,t,i)\mathbb{R}_{unit}(s,o,t,i)\end{aligned} \tag{3.3}$$

where, $\mathbb{R}_{unit}^t = \frac{R_{max}}{A_t}$ is the reward for scanning unit area of task t and $\mathbb{R}_{unit}(s,o,t,i)$ is the unit reward for ith scan by AEOS s in revolution o for task t which includes the penalty. Since the tasks may need multiple scans to be completed successfully, there might be overlaps of areas during the scans. The reward calculation for a task comprising all scans must compute rewards for each overlapping area only once. Let $\tilde{B}^t$ be the n^t overlapped regions in the scans for task t.

$$\tilde{B}^t = \{\tilde{B}^t_1, \tilde{B}^t_2, \dots \tilde{B}^t_{n^t}\} = \bigcap_{(s,o,i)} \tilde{A}(s,o,t,i) \tag{3.4}$$

Since earliness or tardiness penalties are associated with the rewards based on the time of the scan, each scan might collect different rewards for the overlapped areas. The reward accumulated by the nth overlap by AEOS s in revolution o for ith scan can be given by

$$\mathbb{B}^t_n = \tilde{B}^t_n \mathbb{R}_{unit}(s,o,t,i) \tag{3.5}$$

To maximize the reward for a particular task, the scan that collects the maximum reward for the overlapped areas will only be considered. According to the problem definition, since successfully scanning a particular region once is enough to obtain the full reward, the reward collected by all the scans for task t is calculated by finding the summation of the reward for each scanned area only once. This means that except for the scan that gathers the maximum reward, the rest of the $n-1$ overlapped scans for the same area in a task t will need to be ignored in the computation of the reward. The reward to subtract from (3.5) is defined as

$$\begin{aligned} &\sum_{k=1}^{n} \mathbb{B}^t_k - \max(\mathbb{B}^t) \\ &\text{where } \mathbb{B}^t = \{\mathbb{B}^t_1, \mathbb{B}^t_2, ..., \mathbb{B}^t_n\} \end{aligned} \tag{3.6}$$

Thus, the total reward for task t can be calculated as

$$\sum_s \sum_o \sum_i \tilde{R}(s,o,t,i) = \sum_s \sum_o \sum_i \tilde{A}(s,o,t,i)\mathbb{R}_{unit}(s,o,t,i) - \left(\sum_{k=1}^{n} \mathbb{B}_k^t - \max(\mathbb{B}^t)\right) \tag{3.7}$$

$$\text{for } s \in [1, N_s], o \in [1, N_{os}]\, i \in [i_1, i_2, \ldots, i_I]$$

where i_I is the total number of scans by AEOS s in revolution o for task t. The objective function of the considered optimization problem needs to be an aggregation of the accumulated rewards from all the user-specified tasks assigned to all the AEOSs in the mission and is defined as follows

$$\begin{gathered} \underset{\mathcal{D}}{\text{maximize}} \sum_t \sum_s \sum_o x(s,o,t) \sum_i \tilde{R}(s,o,t,i) \\ \text{where } \mathcal{D} \in \{x(s,o,t),\ T_{st}(s,o,t),\ D(s,o,t),\ \Theta(s,o,t),\ \Phi(s,o,t)\} \\ \forall s \in [1, N_s], \forall o \in [1, N_{os}]\, \forall t \in [1, N_t] \end{gathered} \tag{3.8}$$

In order to deal with the complexity and size of the optimization problem defined above, a *multi-stage* optimization framework has been proposed. A flowchart of the described multi-stage optimization problem framework is illustrated in Figure 3.1.

In the first stage, the AEOSs are assigned to the tasks with the consideration of the user-specified time windows. At this stage, the assignment process would involve determining the decision variables associated with the assignment of tasks, their starting times, and their duration, i.e., $x(s,o,t)$, $T_{st}(s,o,t)$ and $D(s,o,t)$ respectively. The assignment of tasks having massive AOIs would involve processing on the basis of the information available prior to the start of the mission, i.e., the AEOS trajectories and the task requests with the corresponding coordinates of

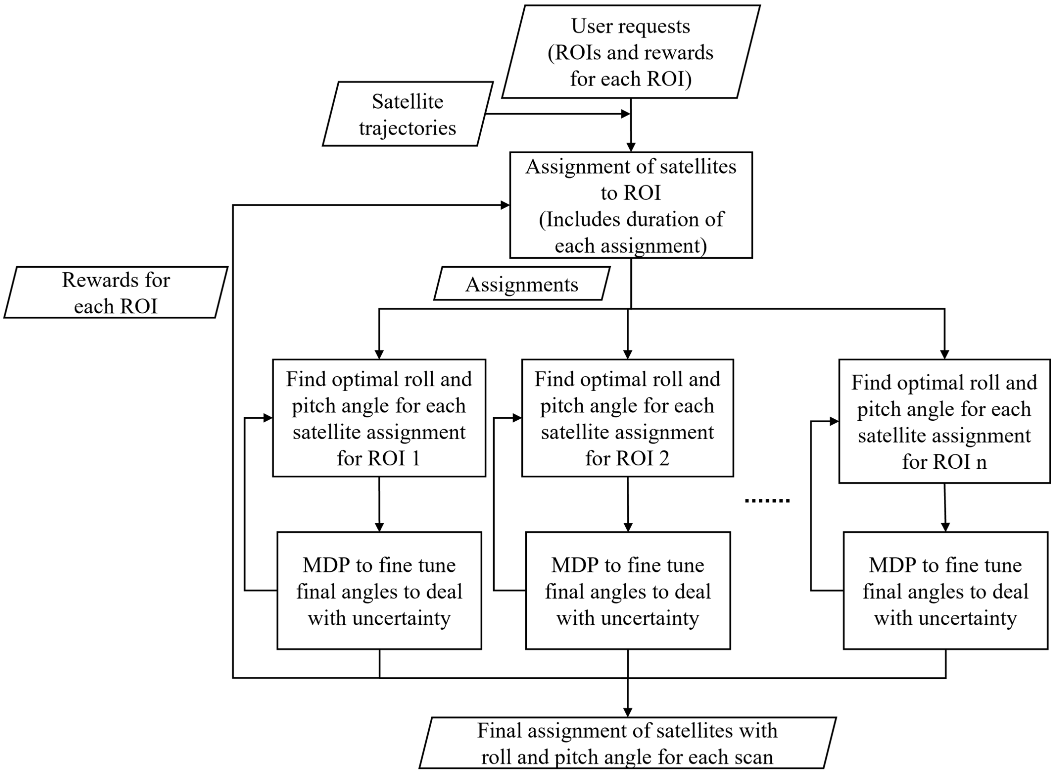


FIGURE 3.1: Flowchart of the multi-stage optimization problem framework

the AOIs. Then, in the second stage, corresponding to the AEOS assignments in the first stage, each AOI is considered individually. Since these tasks are massive, each assigned scan to a single AOI can only scan a portion of the required AOI. The optimal roll and pitch angles of the AEOSs needed to perform the scanning tasks for each assigned scan are calculated at this stage. This means that in the second stage, the optimal roll and pitch angles, $\Theta(s, o, t)$ and $\Phi(s, o, t)$ respectively, are to be determined to cover the maximum area of the specified AOI by the assigned AEOSs. However, the determination of the optimal schedule would involve optimizing the first and the second stages hand-in-hand for several iterations.

As shown in Figure 3.1, at the beginning of the optimization process, the first stage finds a sub-optimal assignment of the satellites, then the second stage finds the roll and pitch angle corresponding to each assignment. The process goes back to the first stage to find a better solution to the assignment problem, and the cycle continues for several iterations. The optimization of $\Theta(s, o, t)$ and $\Phi(s, o, t)$ in the second stage will determine the scanned areas corresponding to the decision variables $x(s, o, t)$, $T_{st}(s, o, t)$ and $D(s, o, t)$ identified in the first stage. This would consequently allow to calculate the reward obtained for the scans corresponding to this combination of decision variables. As these tasks are independent of each other, the second stage of optimization for all the tasks can be done with parallel processing to reduce computational time. The *third* stage deals with the uncertainty originating from the failure of scans. A Markov Decision Process based intermediate stage is introduced to fine-tune the roll and pitch angles for the consecutive assignments, if any previous assignment has failed. Additionally, the role of the third stage will be effective in the identification of roll and pitch angles should any failure of scans occur in an online setting of optimization. The above-mentioned multi-stage optimization framework is illustrated in a detailed manner as follows.

3.3.1 Stage 1: The Optimal Assignment of Tasks to the AEOSs

The assignment of the AEOSs to the tasks in the first stage is optimized by maximizing the reward obtained by N_s AEOSs scanning N_t tasks. In the simplest form, the reward accumulation may be framed in the form of the following objective function,

$$\begin{gathered}\underset{\mathcal{D}_1}{\text{maximize}} \sum_{t=1}^{N_t}\sum_{s=1}^{N_s}\sum_{o=1}^{N_{os}} x(s,o,t)\left(\sum_{i}\left(\prod_{j=1}^{i} C(j)\right)\tilde{R}(s,o,t,i)\right)\\ \text{where } \mathcal{D}_1 \in \{x(s,o,t),\ T_{st}(s,o,t),\ D(s,o,t)\}\end{gathered} \tag{3.9}$$

Here, $x(s,o,t)$ is a binary decision variable that takes the value 1 if AEOS s is assigned to task t in revolution o, otherwise, it takes the value 0. Once an AEOS s has been assigned to a particular task t in revolution o, multiple scans might occur as discussed above. To monitor the number of scans in a particular revolution o by AEOS s for task t, a new binary decision variable $C(j)$ with values of 0 and 1 is introduced, where $j \in [1,i]$. The constraints of the objective function at this stage are defined as follows in equations (3.10) and (3.12). The maximization of the reward $\tilde{R}(s,o,t,i)$ is performed with respect to determining the optimal decision variables $\mathcal{D}_1$.

3.3.1.1 Time Constraint

The schedule needs to consider that a particular AEOS has enough time to finish an assigned scan i before proceeding to the next assignment j. This is incorporated with respect to the user-specified time windows, in terms of the start time and duration of the consecutive scans as follows,

$$t_{st}(s, o, t, j) > t_{st}(s, o, t, i) + d(s, o, t, i) \quad \forall t, i \neq j \tag{3.10}$$

Hence, the constraint (3.10) ensures that any new assignment j for AEOS s in revolution o on task t occurs after the assigned duration for assignment i has passed.

3.3.1.2 Memory Constraint

The onboard memory of AEOS is finite. The memory required for ith scan for task t by satellite s is defined as

$$m(s, o, t, i) = m_{unit} \times d(s, o, t, i) \tag{3.11}$$

When the AEOS passes over the VW of the ground station, it transmits the data from the onboard memory. After each communication with the ground station, the onboard memory is reset. The scans can be performed by the AEOS until the maximum capacity of the onboard memory, m_{max} is reached. The satellite s can only perform a specific number of scans in between the communication to ground station g and the next available ground station $\hat{g}$, which is defined as,

$$\sum_{N_{(s,g,\hat{g})}} m(s,o,t,i) \leq m_{max} \qquad \forall s, g, \hat{g} \neq g \tag{3.12}$$

The value of $N_{(s,g,\hat{g})}$ is calculated from the decision variable $T_{st}(s,o,t)$, which denotes the start time of the scan in the interval between two consecutive communications with the ground station g and $\hat{g}$ for AEOS s.

$$N_{(s,g,\hat{g})} = |T_{st}(s,o,t)| \quad \text{where} \quad (T_g(s,o) + d_g(s,o)) < t_{st}(s,o,t,i) < T_{\hat{g}}(s,o) \tag{3.13}$$

The values of $T_g(s,o)$ and $d_g(s,o)$, which is a function of m_{max}, are predefined.

3.3.2 Stage 2: The Optimal Selection of Roll and Pitch Angles

Now, in the next stage, the optimal roll and pitch angles for each assignment need to be determined in order to maximally cover the AOI corresponding to the task. In the first stage, the assignment of the AEOS to each task is set and passed on to the second stage of the optimization problem. In the context of each task, the combination of roll and pitch angles for each assignment decides the area that will be scanned. The area scanned for each unit time is calculated with the help of trigonometric relationships between the position of the AEOS in its trajectory and the roll and pitch angle of the sensors, as illustrated in Figure 3.2. The AOI corresponding to a single task is represented as the area $PQRS$ in Figure 3.2 on the earth's surface. The dark-shaded region $DEGH$ inside $PQRS$ is the area covered in a single scan for unit time stamp. Let the

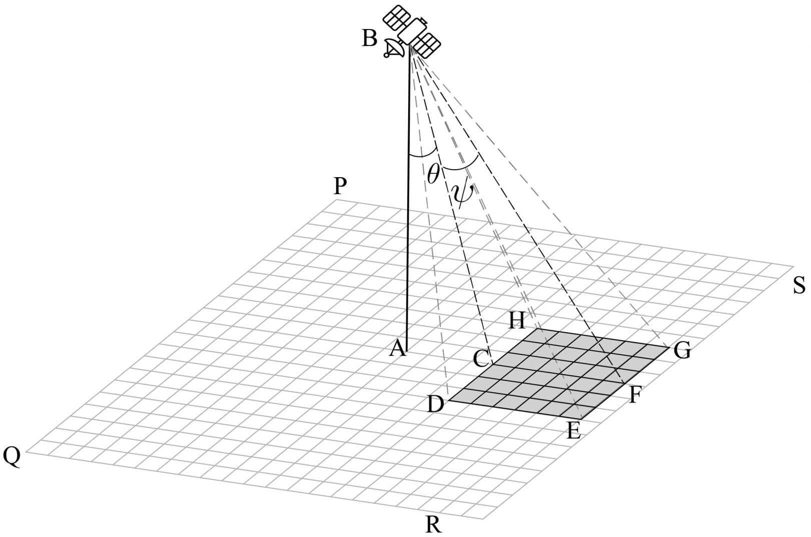


Figure 3.2: Scanned area for unit time stamp arising from each assignment

AEOS be at position B and the projection of the AEOS's position (Spacecraft nadir-point) on the earth's surface be A. The coordinates of B, say, (x, y) are known prior to the mission from the trajectory of the AEOS, which is the same as A. Now, let (θ, ϕ) be the roll and pitch angles of the scan. In this figure, the pitch angle ϕ is considered zero, and the trajectory of the AEOS is parallel to the longitude lines of the earth's surface. Then with the rules of trigonometry, the coordinates of the points C and F can be calculated as $(x, y + h\tan(\theta))$ and $(x, y + h\tan(\theta + \psi))$, where h is the height of the AEOS and ψ is the swath width of the AEOS. Since C is the mid-point between the line segment joining H and D and the swath width of the AEOS is known, the coordinates of H and D can be calculated. Since the length of the line segment HD is equal to CF, the coordinates of H and D can be calculated as $\Big(\Big(x + \frac{h}{2}(\tan(\theta + \psi) - \tan\theta)\Big), (y + h\tan(\theta))\Big)$ and $\Big(\Big(x - \frac{h}{2}(\tan(\theta + \psi) - \tan\theta)\Big), (y + h\tan(\theta))\Big)$ respectively. In a similar manner, the coordinates of G and E are $\Big(x + \frac{h}{2}\left[\tan(\theta + \psi) - \tan\theta\right], y + h\tan(\theta + \psi)\Big)$ and

$\left(\left(x - \frac{h}{2}(\tan(\theta + \psi) - \tan\theta)\right), (y + h\tan(\theta + \psi))\right)$ respectively. Then, the area covered by a unit timestamp $DEGH$ can be determined. This illustrates an instance of the determination of the decision variable $\tilde{A}(s, o, t, i)$ corresponding to the area covered in the ith scan of task t by AEOS s with revolution o. At this stage, the following objective function maximizes the total area covered for task t,

$$\begin{aligned} &\underset{\mathcal{D}_2}{\text{maximize}} \bigcup_{s,o,i} \tilde{A}(s, o, t, i) \\ &\text{where } \mathcal{D}_2 \in \{\Theta(s, o, t),\ \Phi(s, o, t)\} \end{aligned} \tag{3.14}$$

3.3.2.1 Roll and Pitch Angles Constraints

According to the specifications of the AEOS, the maximum possible roll and pitch angles for each scan are defined. During the scan, these decision variables need to follow the constraints stated below,

$$\theta_t^i \leq \Theta_{max_s} \tag{3.15}$$

$$\phi_t^i \leq \Phi_{max_s} \tag{3.16}$$

Equations (3.15, 3.16) ensure that the assigned roll and pitch angles are within the maximum limits of the specifications of the AEOS. The above-defined objective function and constraints in the second stage allow to find the maximum area coverage by optimizing the roll and pitch angles. As discussed in equations (3.4) to (3.7), the rewards accumulated during the overlapped region are discarded. Since the first and second stage of optimization is performed simultaneously for multiple iterations, the reward obtained by scanning the region $\tilde{A}(s, o, t, i)$ is given as,

$$\tilde{R}(s,o,t,i) = \tilde{A}(s,o,t,i)\mathbb{R}_{unit}(s,o,t,i) - \left(\sum_{k=1}^{n} \mathbb{B}_k^t - \max(\mathbb{B}^t)\right) \tag{3.17}$$

It should be noted that the reward $R(s,o,t,i)$ is the same as in equation (3.9) in stage 1 of the optimization framework. Since each task is independent of the other, maximizing the reward corresponding to each task can be calculated with parallel processing.

3.3.3 Stage 3: Markov Decision Process to handle uncertainty

Due to unpredictable cloud coverage, sensor and equipment malfunctions and other factors, failure of scans might occur. For handling the uncertainties arising due to these unprecedented reasons, an MDP-based approach is proposed as an additional step in the multi-stage optimization framework. Since these tasks correspond to massive AOIs, multiple scans are needed to cover the whole region. Failure of a single scan would impact the coverage of the whole task corresponding to the existing assignment. A re-allocation of scanning areas for the upcoming assignments in the mission horizon might be helpful to append some portions of the missed region of the failed scan. This is achieved through the determination of an updated optimal choice of roll and pitch angles of the consequent assignment to ensure maximum coverage, including portions of missed regions, and hence maximize the reward.

3.3.3.1 State space ($\mathcal{S}$)

The sample state space for the MDP is defined as follows,

- ν – Area to be scanned in the next assignment

- w – Weather prediction for the next scan

Here,

$$\nu \in \mathcal{V}_t, \text{ where } \mathcal{V}_t = \bigcup_{\forall\ s,\ o,\ i} \tilde{A}(s, o, t, i) \tag{3.18}$$

The set of assignments corresponding to the areas $\tilde{A}(s, o, t, i)$ for task t is defined as $\mathcal{V}_t$. During the mission horizon, the AEOSs get updated real-time information about the weather. Based on the current weather, the state variable w is defined as 0 or 1, where 0 indicates bad weather conditions and 1 indicates good weather conditions. A multi-tier weather condition state variable can also be used instead of the two-tier$(0, 1)$ but that will increment the number of states and the complexity.

3.3.3.2 Action space ($\mathcal{A}$)

The possible actions for the MDP, depending upon the current state, are given as follows,

- a_s – Skip the next scan
- a_p – Proceed to the next scan

By the action a_s, the AEOS will skip the next scheduled scan and preserve the resources. The action a_p will perform the scan and depreciate resources irrespective of the outcome.

3.3.3.3 Transition Probability ($\mathcal{T}$)

The transition probabilities for the actions are dependent on weather conditions, sensor failure, and other factors. When the weather condition is favorable ($w =$

1), the probability of success (ξ_g) for a scan is higher than when the weather is unfavorable, ($w = 0$). The probability of success when the weather condition is not favorable, is defined as ξ_b. Sensor failures and other factors may still result in a scan failure even if the weather is in favour. The probability condition is reversed when the weather condition is adverse ($w = 0$). The transition probability for having a successful or failed scan can be defined as:

$$\begin{aligned}
\Pr(Success|a_p, w = 1) &= \xi_g \\
\Pr(Failure|a_p, w = 1) &= (1 - \xi_g) \\
\Pr(Success|a_p, w = 0) &= \xi_b \\
\Pr(Failure|a_p, w = 0) &= (1 - \xi_b) \\
\Pr(Failure|a_s, w = *) &= 1 \qquad (3.19)
\end{aligned}$$

The probability of the weather prediction is defined as:

$$\begin{aligned}
\Pr(w = 1|a_*, w = *) &= w_p \\
\Pr(w = 0|a_*, w = *) &= (1 - w_p) \qquad (3.20)
\end{aligned}$$

where $0 \leq \xi_g, \xi_b \leq 1$ is the success rate of a scan and $0 \leq w_p \leq 1$ is the weather prediction at each state, a_* denotes either of the action and $w = *$ denotes either state condition for w. The final transitional probability is calculated by the product of (3.19) and (3.20).

3.3.3.4 Reward function($\mathcal{R}$)

The reward function for the MDP includes the following factors,

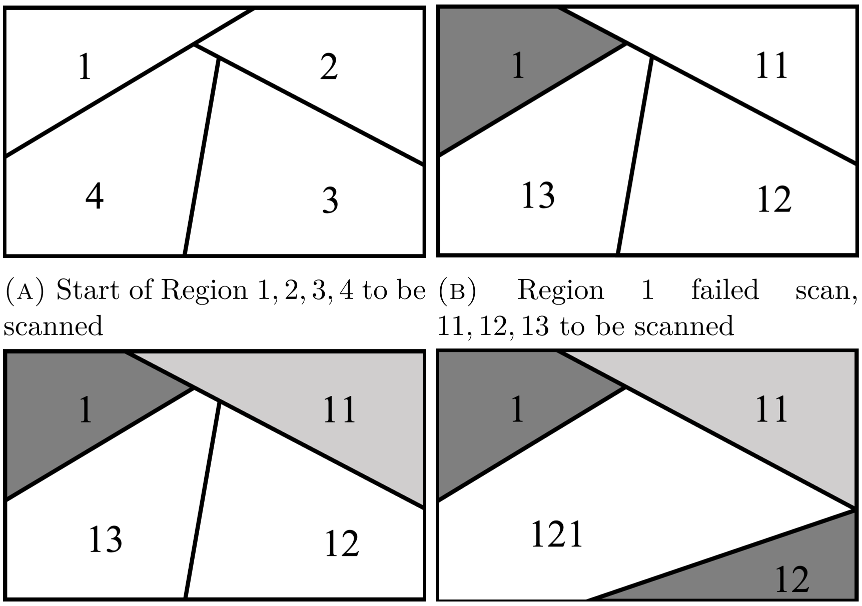
- reward for completing the current scan
- no reward for skipping the scan
- penalty for utilizing the resource

The penalty for using the resources is applied every time a scan is performed, irrespective of the success or failure of the scan.

$$R^t = \begin{cases} \tilde{R}(s,o,t,i) - p \times m(s,o,t,i) & \text{if scan is successful} \\ -p \times m(s,o,t,i) & \text{if scan is unsuccessful} \\ 0 & \text{if scan is skipped} \end{cases} \tag{3.21}$$

where R^t is the reward for the state transition. Here, p is the penalty for utilizing the unit resources. The penalty is chosen by the user. The AEOSs can conserve resources by skipping a scan. In this work, the utilization of the reserved resource is not considered but kept for future work.

A sample state space with actions for the example from subfigure 3.3 for the above MDP model is shown in Figure 3.4. The subfigure 3.3a illustrates an instance of area distribution for scans obtained by optimization of roll and pitch angles in the second stage. It is shown that there are four scans to be performed in the considered task AOI. The areas are named 1, 2, 3 and 4, corresponding to the order of the scan in the mission horizon. This area distribution will yield the maximum reward if all these four scans are successful as calculated in (3.9) and (3.17). However, as illustrated in subfigure 3.3b, if scan 1 is unsuccessful, the second stage of the optimization is called again, and the roll and pitch angles are



(A) Start of Region 1, 2, 3, 4 to be scanned

(B) Region 1 failed scan, 11, 12, 13 to be scanned

(C) Region 11 success,12, 13 to be scanned

(D) Region 12 failed scan,121 to be scanned

FIGURE 3.3: The re-allocation of the areas corresponding to assignments arising from unsuccessful scans

recalculated for the remaining three assignments to maximize the area coverage. In subfigure 3.3b, the change in the area distribution after the recalculation and the updated numbering of the areas are illustrated. The adopted rule of numbering the re-allocated areas includes the index of the unsuccessful scan followed by the order of the scan according to the mission horizon. Since the scan corresponding to the area 11 is successful, no further reallocation of the area is needed for the remaining scans 12 and 13, as shown in subfigure 3.3c. Further in the mission horizon, when the scan corresponding to area 12 is unsuccessful, the second stage is called again to find an optimal roll and pitch to maximize the area corresponding to scan 121, illustrated by subfigure 3.3d. Figure 3.4 shows some of the states for the example above. The next section illustrates the solution methodology for the formulated problem in this section.

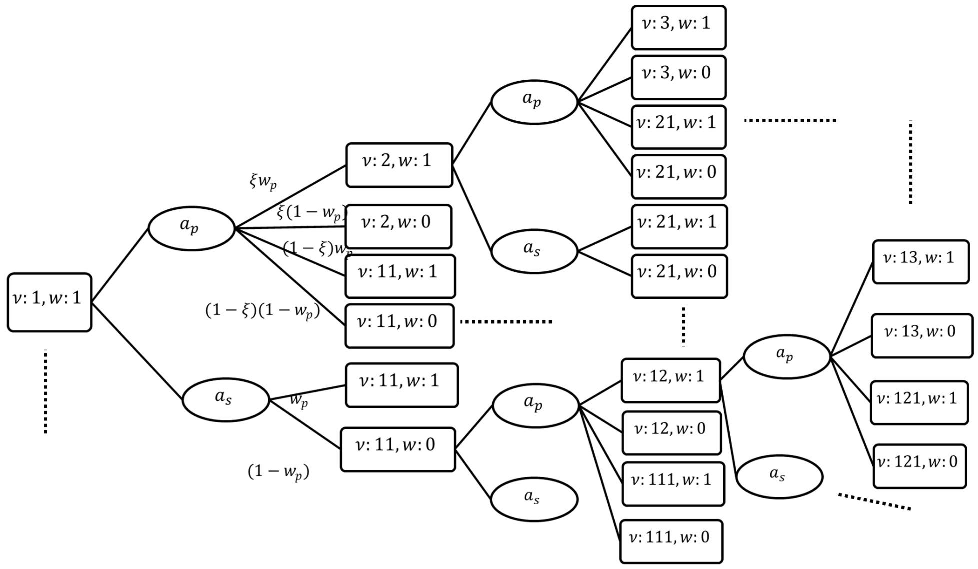


Figure 3.4: Tree diagram for the states of the MDP for instance illustrated in Figure 3.3

3.4 Solution Methodology

In the current section, solution techniques for the proposed optimization model are illustrated. As discussed in section 3.3, a three-stage optimization model has been developed with the help of a randomized heuristic using an elitist genetic algorithm technique (in the first stage), constrained non-linear optimization solver (in the second stage) coupled with MDP solution strategy (in the third stage). The randomized heuristic which is an adaptation of the solution strategy in (Chatterjee and Tharmarasa; 2022) is described in the following subsection. For the second stage, any constrained non-linear optimization algorithm will be suitable, however, for this paper, the interior point method has been used in the simulations (Byrd et al.; 1999). The third stage of the problem is solved using the value iteration technique (Tharmarasa, Chatterjee, Wang, Kirubarajan, Berger and Florea; 2019). Prior to the mission, the third stage fine-tunes the schedule from the first and second stages. In order to reduce the computational load, the third stage can be started after a few initial generations of the first and second stages. A comparison of this is shown in the simulation section below.

Elitist Mixed Coded Scheduling (EMCS)

EMCS is based on a modified elitist genetic algorithm (Kim and Ellis Jr; 2008) with binary and continuous variables together in the structure of the chromosome. EMCS considers an initial population of candidate solutions, which are random schedules of all the AEOSs performing the tasks across the whole mission horizon. Our goal is to find the near-optimal schedule, that is near-optimal chromosome from the population. The algorithm uses crossover and mutation of schedules to create offspring, and an elitist selection operator is used to select from a pool of offspring schedules. The chromosomes are sorted based on the reward accumulated for the corresponding schedule. In each generation, the best half of the population is chosen for creating the next generation with the help of mutation, crossover and elitist selection operators. The structure of the EMCS is described in Algorithm 3. Operations of the algorithm are detailed below:

Where λ (population size) and *threshold* are both user-defined and based on the complexity of the problem.

3.4.1 Preprocessing

A set of preprocessing operations are performed to make the algorithm computationally efficient. A task may not be present in the VTW of an AEOS in all of its orbits of the AEOS trajectory. Depending on the trajectory of the AEOS and the

Algorithm 3: Elitist Mixed Coded Scheduling

```
Set population of schedules, $\mathcal{P}_0 = \varnothing$;
while population size, $|\mathcal{P}_0| \leq \lambda$ do
    Allocate a task randomly along with the task start time and the
      duration to each AEOS in each orbit to generate random chromosome
      $p$;
    The task start time and the duration corresponding to each
      assignment are selected as a random variable that falls inside the
      VTW of each exposure. $\mathcal{P}_0 = \mathcal{P}_0 \cup p$;
$Gen = 0$;
while Stopping criteria not reached do
    if $Gen \leq threshold$ then
        Sort population $\mathcal{P}_{Gen}$ by reward accumulated only in second stage;
    else
        Sort population $\mathcal{P}_{Gen}$ by reward accumulated with second stage;
    Select $\lambda/2$ best chromosomes from population, $\mathcal{P}_{Gen}^{\lambda/2}$;
    for $|\mathcal{P}_{Gen+1}| \leq \lambda$ do
        Select 2 parents $p_1$, $p_2$ randomly from $\mathcal{P}_{Gen}^{\lambda/2}$;
        Obtain children $c_1$, $c_2$ by crossover on $p_1$, $p_2$;
        Mutate $c_1$, $c_2$ to obtain $cm_1$, $cm_2$;
        Check feasibility of $c_1, c_2, cm_1, cm_2$ ;
        Add best feasible individual to $\mathcal{P}_{Gen+1}$;
    $Gen = Gen + 1$;
```

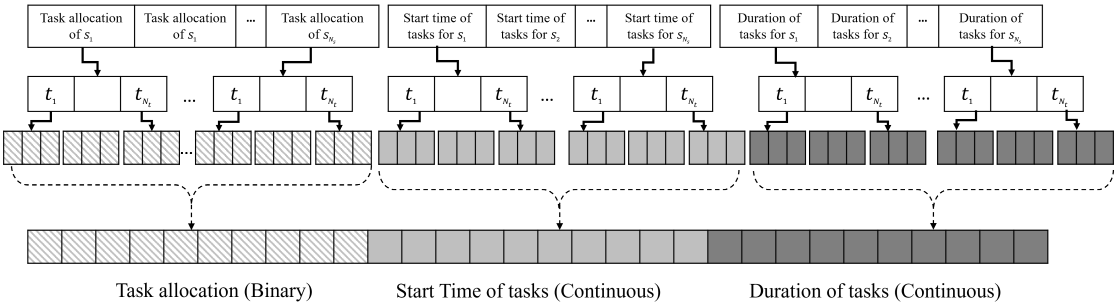

Task allocation of s_1
Task allocation of s_1
...
Task allocation of s_{N_s}
Start time of tasks for s_1
Start time of tasks for s_2
...
Start time of tasks for s_{N_s}
Duration of tasks for s_1
Duration of tasks for s_2
...
Duration of tasks for s_{N_s}
t_1
t_{N_t}
...
Task allocation (Binary)
Start Time of tasks (Continuous)
Duration of tasks (Continuous)

FIGURE 3.5: Structure of chromosome for the algorithms

specifications (maximum roll and pitch angle) of the AEOS, the orbits in which a task is exposed to the AEOS are calculated. This preprocessing helps to remove the redundant orbits from the search, where a task is not exposed to the AEOS. This makes the algorithm efficient by reducing the size of the chromosomes.

3.4.2 Structure of Chromosome

Each chromosome has three parts as shown in Figure 3.5. The dashed section represents the task assignment, the light grey section represents the start time of the operations, and the darker grey section represents the duration of the operation for the AEOSs. The respective schedules of N_s AEOSs are juxtaposed to create the chromosome. Since the exposed orbits for each AEOS corresponding to each task calculated in the preprocessing stage are not equal, the number of bits corresponding to the task allocation of each AEOS is different. A schedule of an AEOS is designed as a combination of bits containing binary and continuous numbers. The possibility of occurrence of the scan for a task on an exposed orbit $x(s, o, t)$ is the binary bits stored at the beginning of the chromosome. The length of the binary bits is the same as the total number of exposed orbits for all the tasks. The start-time of each corresponding scan $t^{st}(s, o, t, i)$ and the duration $d(s, o, t, i)$ are stored in the continuous bits, following the binary bits. The dotted line shows how the sections are adjoined to create the whole chromosome. The total length of the bits containing start-time and duration are the same as the length of the binary bits as illustrated in Figure 3.5.

3.5 Simulation

In this section, the above-mentioned methodology has been illustrated with experiments on simulated scenarios. The experiments are carried out with MATLAB r2020a in an Intel i7, 10th generation processor with 16GB of RAM. The simulated scenarios corresponding to the number of tasks are described in the following subsections in detail. There are five identical AEOSs in the mission, and the number of AEOSs is kept the same for both small-scale and large-scale. The time horizon for the mission to be planned has been set to 12 hours. The user-requested tasks that need to be scheduled are known prior to the start of the mission. The optimization process for the assignment of tasks to the AEOSs is performed prior to the start of the mission. The trajectories of the AEOSs are predefined. From the trajectory of the AEOSs, it is calculated that one revolution takes around 1 hour 20 minutes for each AEOS. The maximum roll and pitch angles for each AEOS are set to 50° and 60°, respectively. The swath width of the satellites is considered 80kms. Table 4.1 shows the orbital elements for the 5 satellites used in the simulation in the two-line element (TLE) format.

Table 3.1: Orbital elements of the satellites in TLE format

Sat 1	1 99999U 15182.00000000 .00000125 00000-0 12143-4 0 00008 2 99999 097.7544 188.7205 0002363 359.8062 142.7546 14.92589026000017
Sat 2	1 99999U 14350.00034722 .00000061 00000-0 59555-5 0 00000 2 99999 097.7404 354.5712 0012530 325.4255 209.1276 14.92590498000011
Sat 3	1 99999U 14350.00034722 -.00000135 00000-0 -13100-4 0 00009 2 99999 097.7404 354.5711 0012490 325.3906 089.1576 14.92590756000013
Sat 4	1 99999U 14350.00034722 -.00000125 00000-0 -10109-4 0 00009 2 99999 097.7504 254.5918 0012630 345.4306 109.1476 14.92590156000013
Sat 5	1 99999U 15230.00022326 -.0000075 00000-0 -11105-4 0 00007 2 99999 097.7404 178.3841 0012490 327.3726 136.1735 14.92590756000013

3.5.1 Small-scale scenario

In the small-scale scenario, ten user-requested tasks are considered, and the task parameters are described in Table 3.2. In Table 3.2, the first column represents the task ID, and the second column represents the corresponding maximum achievable reward. The scanning tasks are set up as quadrilateral regions, which cannot be scanned in a single attempt. The third to sixth columns of table 3.2 represent the coordinates of the chosen task vertices. The eighth and ninth columns represent the early penalty and the late penalty for each task. The last two columns represent the user-specified time requirement for each task referring to the start of the mission as 00:00:00.

By considering the maximum roll and pitch angles, the exposure of each task to a particular AEOS on its revolution is calculated. Prior to the multi-stage optimization, a pre-processing of the simulated data is done to eliminate redundant revolutions in which no tasks are exposed to the AEOS. Table 3.3 illustrates the number of exposures of each task for each AEOS. It should be noted that, due to the positional advantages of tasks $8, 9$ and 10 with respect to the AEOS trajectories

TABLE 3.2: Specifications of the considered tasks for the small-scale scenario

Task	**Reward**	**Point 1**	**Point 2**	**Point 3**	**Point 4**	**Total area (km^2)**	**Early penalty**	**Late penalty**	**Start time**	**End time**
1	10	(45.5, -75.5)	(46.5,-76.5)	(46.5, -73.5)	(45.5,-73.5)	21600	0.01	0.01	00:44:10	00:54:20
2	10	(45.5, -76)	(46.5,-77)	(43.5, -73)	(42.5,-77)	54364	0.01	0.01	03:34:00	03:57:50
3	10	(45.5, -76.5)	(46.5,-77.5)	(44.5, -75)	(41,-77.5)	36914	0.01	0.01	00:05:50	00:17:10
4	10	(43.75, -79.2)	(44.75,-80.2)	(43.5, -76.5)	(42.5,-79)	45698	0.01	0.01	04:36:10	05:52:30
5	25	(43.75, -79.6)	(44.75,-80.6)	(41.5, -89)	(47.5,-85)	15840	0.02	0.02	00:11:20	00:22:30
6	10	(55, -135)	(56,-136)	(55.5, -145)	(57.5,-125)	107610	0.01	0.01	06:16:40	09:03:20
7	17	(53, -130)	(54, -131)	(61.5 ,-159)	(57.5,-145)	186200	0.02	0.02	07:45:00	10:40:00
8	11	(62, -67)	(63,-68)	(61.5, -59)	(67.5,-65)	108560	0.01	0.01	05:00:50	07:21:40
9	15	(66, -76)	(67, -77)	(61.5, -79)	(67.5, -71)	58608	0.02	0.02	10:11:40	11:26:40
10	20	(67, -58)	(68, -59)	(51.5, -59)	(57.5, -55)	248670	0.02	0.02	06:43:20	10:45:00

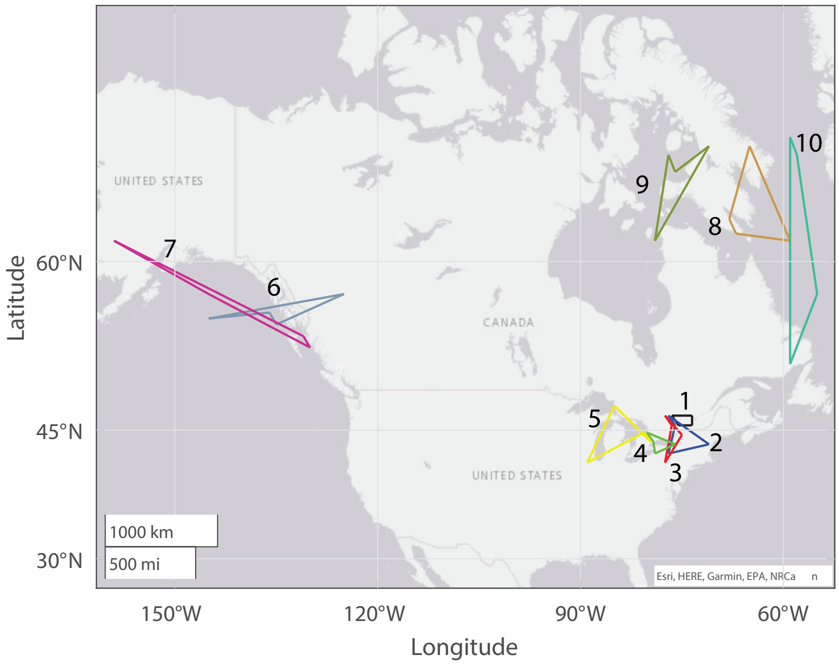


FIGURE 3.6: Locations of the tasks

considered, these tasks are exposed to the AEOSs in a higher number of orbits than the other tasks.

In this simulation, the AEOSs periodically communicate with 6 ground stations. Table 3.4 shows the locations of the considered ground stations in terms of latitude and longitude. Prior to the mission, the number of possible communications of the AEOSs with the ground stations based on the VTW of the ground stations is

TABLE 3.3: Task-wise exposure details of five AEOSs

Tasks →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Sat 1	1	1	1	1	1	1	2	3	3	4
Sat 2	1	1	1	1	1	1	1	4	3	4
Sat 3	2	2	2	1	1	1	1	3	4	4
Sat 4	1	1	1	1	1	1	2	3	3	4
Sat 5	1	1	1	1	1	1	2	3	3	4

calculated in a similar manner of preprocessing as done for task exposures.

Table 3.4: Ground station locations

Name	Latitude (deg)	Longitude (deg)
Ground station 1	45.58	-75.81
Ground station 2	68.31	-133.5
Ground station 3	53.43	-105.18
Ground station 4	-15.00	30.00
Ground station 5	28.59	77.15
Ground station 6	-7.00	79.00

Table 3.5 shows the number of times the AEOSs are exposed to a particular task AOI combined from table 3.3. This table further shows how many times each task has been scanned and the reward collected by all the AEOSs for each task. It is noticeable from table 3.5 that for tasks $6 - 10$ the number of scans is higher, and AEOSs have collected the full reward from them. For task $1 - 5$ because of their close positioning (can be seen in figure 3.6), the number of scans by all the AEOSs is less, and they have not received the full reward.

Table 3.6 demonstrates the effects of redistributing the roll and pitch angle by using the MDP when a scan is unsuccessful. For task 1, all the scans are successful, so the MDP did not affect the total reward accumulated. For task 2, only 2 out of the 3 scans were successful. The MDP redistributed the area, and it yielded more rewards. A similar scenario is observed for tasks 6 and 8.

The success rate of a scan for the AEOSs determines the total reward collected by the AEOSs during the mission. Table 3.7 shows that using the MDP to reallocate the areas to be scanned yields higher reward collected. The value of ξ_b is kept at 0.1 and $w_p = 0.9$. When the ξ_g value is low, which means the probability of a successful scan is low, the inclusion of MDP helps to gather more rewards.

Table 3.5: Total number of scans performed by all the AEOSs

	Number of Exposure	**Scans done**	**Max reward**	**Reward collected**
T1	6	3	10	8.3
T2	6	3	10	6.65
T3	6	1	10	4.15
T4	5	2	10	7.38
T5	5	2	25	9.36
T6	5	4	10	10
T7	8	5	17	17
T8	16	6	11	11
T9	16	6	15	15
T10	20	7	20	20

In the simulation, it took an average of 380 generations of the first stage to reach convergence. Table 3.8 compares how the total reward accumulated changes based on when the 3rd stage is introduced to fine-tune the result. It is noted that the processing time is higher when the 3rd stage is used from the earlier part of the solution but yields more reward than when introduced in the later part.

Table 3.9 compares the reward collected by the proposed method in this work and the conventional strip method. From the table, it can be observed that in most of the tasks, the proposed method of assigning roll and pitch angel to create strips that are not parallel to the trajectory of the AEOSs collects more reward than the conventional strip method (Xu et al.; 2018). It is also shown in table 3.9, the memory consumed by the strip-based method is more than the proposed method. Figure 3.7(a) shows that in the conventional strip method how the strips are formed parallel to the trajectory of the AEOSs, which are marked as the dotted line. The green polygon is the area of task 4. Figure 3.7(b) shows which strips

Table 3.6: Online performance with MDP

	Scans Assigned	**Successful Scan**	**Max reward**	**Reward collected with MDP**	**Reward collected without MDP**
T1	3	3	10	8.3	8.3
T2	3	2	10	6.65	5.72
T3	1	1	10	4.15	4.15
T4	2	2	10	7.38	7.38
T5	2	2	25	9.36	9.36
T6	4	3	10	8.54	7.86
T7	5	5	17	17	17
T8	6	5	11	9.67	8.65
T9	6	6	15	15	15
T10	7	7	20	20	20
Total			**138**	**106.05**	**103.42**

TABLE 3.7: Performance comparison for different rates of ξ_g with $\xi_b = 0.1$

ξ_g	**Total reward collected with MDP**	**Total reward collected without MDP**
1	109.68	109.68
0.9	106.05	103.42
0.8	103.67	97.35
0.7	100.94	94.28
0.6	97.17	90.47
0.5	92.88	84.53

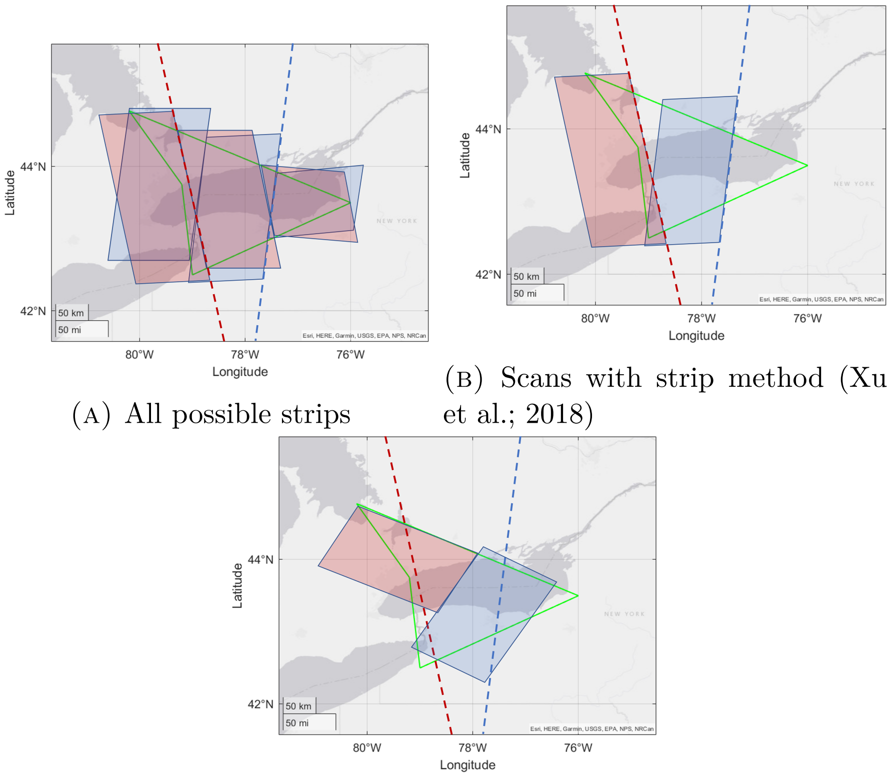
TABLE 3.8: Performance comparison for the use of MDP in the simulation

	With MDP from the beginning	**With MDP starting at 100th generation**	**With MDP starting at 200th generation**	**Without MDP**
Reward collected	109.76	109.04	105.24	103.42
Normalized processing time	1	0.89	0.81	0.72

are actually scanned during the mission with the strip method. The duration of a scan will depend on the task size, satellite trajectory, and the swath width of the satellite. Figure 3.7(c) shows that with the proposed methodology, the area scanned by the two AEOSs is not parallel to the trajectory, and they cover more area than the strip method as shown in Figure 3.7(b).

3.5.2 Large-scale scenario

For a large-scale scenario, a significantly larger task allocation problem with 400 tasks is considered. This section of the simulation provides insight into the task allocation when the number of tasks increases significantly. These tasks have rewards uniformly distributed between 10 and 30. The areas of these tasks are uniformly distributed throughout the map shown in Figure 3.6. The considered



(A) All possible strips

(B) Scans with strip method (Xu et al.; 2018)

(C) Scans with proposed approach

FIGURE 3.7: Comparison of Strip method vs. proposed method for task 4

AEOS parameters are kept the same as in the small-scale scenario. The structure of the chromosome is modified according to the exposures of the tasks for all of the five AEOSs. The ground station specifications are also kept the same as the small-scale scenario. Figure 3.8 shows the comparison between the proposed methodologies and the strip-based method. It shows that with the proposed method, the total reward accumulated is much larger than the strip-based method.

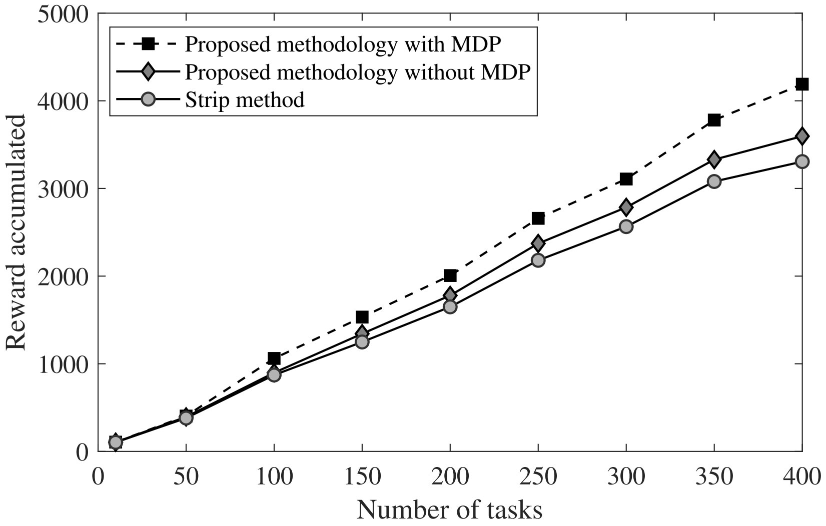


Figure 3.8: Reward comparison for the proposed method and the strip method in large-scale scenario

3.6 Conclusion

In this work, a three-stage AEOS scheduling model has been proposed to find the optimal assignment to task regions that have large areas. The model includes finding the optimal time and duration for each scan by the AEOSs along with the optimal roll and pitch angles for each scan. The proposed methodology minimizes the region overlapping between the scans, which is a drawback of strip-based task segregation for larger task regions. In both small and large-scale scenarios, this has been illustrated with simulation results. The inclusion of MDP in dealing with task failure uncertainty provides superior performance in terms of reward accumulation. This work has future scopes of research considering an application-oriented fixed quantity of necessary overlapping in scans to account for compensation errors and the arrival of new tasks during the mission.

Bibliography

Berger, J. (2016). Mult-satellite intelligence collection scheduling, *DRCD Valcartier Research Centre, Dec* .

Byrd, R. H., Hribar, M. E. and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming, *SIAM Journal on Optimization* **9**(4): 877–900.

Chatterjee, A. and Tharmarasa, R. (2022). Reward factor-based multiple agile satellites scheduling with energy and memory constraints, *IEEE Transactions on Aerospace and Electronic Systems* **58**(4): 3090–3103.

Cordeau, J.-F. and Laporte, G. (2005). Maximizing the value of an earth observation satellite orbit, *Journal of the Operational Research Society* **56**(8): 962–968.

Eddy, D. and Kochenderfer, M. (2020). Markov decision processes for multi-objective satellite task planning, *2020 IEEE Aerospace Conference*, IEEE, pp. 1–12.

Globus, A., Crawford, J., Lohn, J. and Pryor, A. (2004). A comparison of techniques for scheduling earth-observing satellites, *Applications of Artificial Intelligence Conference*.

Gu, Y., Han, C., Chen, Y., Liu, S. and Wang, X. (2022). Large region targets observation scheduling by multiple satellites using resampling particle swarm optimization, *IEEE Transactions on Aerospace and Electronic Systems* **59**(2): 1800–1815.

Habet, D., Vasquez, M. and Vimont, Y. (2010). Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite, *Computational optimization and applications* **47**: 307–333.

Hall, N. G. and Magazine, M. J. (1994). Maximizing the value of a space mission, *European journal of operational research* **78**(2): 224–241.

Han, C., Gu, Y., Wu, G. and Wang, X. (2022). Simulated annealing-based heuristic for multiple agile satellites scheduling under cloud coverage uncertainty, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(5): 2863–2874.

He, L., Liu, X., Laporte, G., Chen, Y. and Chen, Y. (2018). An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling, *Computers & Operations Research* **100**: 12–25.

Ju, J. and Roy, D. P. (2008). The availability of cloud-free landsat ETM+ data over the conterminous united states and globally, *Remote Sensing of Environment* **112**(3): 1196–1211.

Kim, J.-L. and Ellis Jr, R. D. (2008). Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling, *Journal of construction engineering and management* **134**(11): 904–913.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M. and Bataille, N. (2002). Selecting and scheduling observations of agile satellites, *Aerospace Science and Technology* **6**(5): 367–381.

Lin, C.-H., Tsai, P.-H., Lai, K.-H. and Chen, J.-Y. (2012). Cloud removal from multitemporal satellite images using information cloning, *IEEE transactions on geoscience and remote sensing* **51**(1): 232–241.

Liu, X., Laporte, G., Chen, Y. and He, R. (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time, *Computers & Operations Research* **86**: 41–53.

Niu, X., Tang, H. and Wu, L. (2018). Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm, *International journal of disaster risk reduction* **28**: 813–825.

Peng, G., Song, G., He, Y., Yu, J., Xiang, S., Xing, L. and Vansteenwegen, P. (2020). Solving the agile earth observation satellite scheduling problem with time-dependent transition times, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52**(3): 1614–1625.

Renjie, H., Baocun, B., Yingwu, C. and Yuejin, T. (2008). Multi-satellite mission planning for environmental and disaster monitoring satellite system, *SpaceOps 2008 conference*, p. 3488.

Tharmarasa, R., Chatterjee, A., Wang, Y., Kirubarajan, T., Berger, J. and Florea, M. C. (2019). Closed-loop multi-satellite scheduling based on hierarchical MDP, *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–7.

Tharmarasa, R., Kirubarajan, T., Berger, J. and Florea, M. C. (2019). Mixed open-and-closed loop satellite task planning, *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–8.

Wang, X., Song, G., Leus, R. and Han, C. (2019). Robust earth observation satellite scheduling with uncertainty of cloud coverage, *IEEE Transactions on Aerospace and Electronic Systems* **56**(3): 2450–2461.

Wang, X.-W., Chen, Z. and Han, C. (2016). Scheduling for single agile satellite, redundant targets problem using complex networks theory, *Chaos, Solitons & Fractals* **83**: 125–132.

Wang, X., Wu, G., Xing, L. and Pedrycz, W. (2020). Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions, *IEEE Systems Journal* .

Wen, Z., Li, L., Song, J., Zhang, S. and Hu, H. (2023). Scheduling single-satellite observation and transmission tasks by using hybrid actor-critic reinforcement learning, *Advances in Space Research* **71**(9): 3883–3896.

Wolfe, W. J. and Sorensen, S. E. (2000). Three scheduling algorithms applied to the earth observing systems domain, *Management Science* **46**(1): 148–166.

Wu, J., Song, B., Zhang, G., Ou, J., Chen, Y., Yao, F., He, L. and Xing, L. (2022). A data-driven improved genetic algorithm for agile earth observation satellite scheduling with time-dependent transition time, *Computers & Industrial Engineering* **174**: 108823.

Xu, Y., Liu, X., He, R. and Chen, Y. (2020). Multi-satellite scheduling framework and algorithm for very large area observation, *Acta Astronautica* **167**: 93–107.

Xu, Y., Liu, X., He, R., Chen, Y. and Chen, Y. (2018). Multi-satellite scheduling framework and algorithm for very large area observation, *2018 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, pp. 1–8.

Zhai, X., Niu, X., Tang, H., Wu, L. and Shen, Y. (2015). Robust satellite scheduling approach for dynamic emergency tasks, *Mathematical Problems in Engineering* **2015**.

Table 3.9: Performance comparison between the strip method and the proposed method

	Max reward	**Reward collected with strips**	**Reward collected with proposed method**	**Memory consumed with strips**	**Memory consumed with proposed method**
T1	10	6.39	8.3	59	48
T2	10	5.48	6.65	46	35
T3	10	3.26	4.15	31	20
T4	10	6.11	7.38	55	47
T5	25	8.23	9.36	87	72
T6	10	6.92	8.54	57	50
T7	17	17	17	139	117
T8	11	8.32	9.67	84	74
T9	15	15	15	126	112
T10	20	20	20	157	129
Total	**138**	**96.71**	**106.05**	**841**	**704**

Chapter 4

Learning-Based Predictive Scheduling for Multiple Agile Satellites with Task Arrivals During Mission

The content of this chapter is submitted to IEEE Transactions on Aerospace and Electronic Systems

Chatterjee, Abhijit and Tharmarasa, Ratnasingham (2023) Learning-based predictive scheduling for multiple agile satellites with new task arrivals during mission, *Submitted to IEEE Transactions on Aerospace and Electronic Systems.*

Learning-Based Predictive Scheduling for Multiple Agile Satellites with New Task Arrivals During Mission

Abstract

In an Agile Earth Observation Satellite (AEOS) mission, the user-requested tasks arrive prior to the start of the mission, as well as, during the mission. Although the constraints on computational costs and available time to schedule those tasks that have arrived prior to the mission are more relaxed, these constraints are much stricter for scheduling new task arrivals during the mission. Complete rescheduling of the tasks during the mission will also create havoc by disrupting the original mission schedule that is generated prior to the mission. A four-phase online-offline predictive schedule strategy, where historical data is used to predict the pattern of the newly arrived tasks, is proposed in this work to include the newly arrived tasks with minimum disruption and computational cost. The predictive schedule created with dummy tasks creates idle spaces in the original mission schedule while minimizing disruption. During the mission, newly arrived tasks are assigned to the idle spaces corresponding to the dummy tasks with the actual new tasks using a fast polynomial-time assignment algorithm. Some of the tasks which can not be assigned are used for rescheduling. This results in a more efficient and less

disruptive schedule than performing complete rescheduling multiple times when new tasks arrive. This combination of offline-online strategy is illustrated with simulations.

Keywords: *Agile satellite scheduling, predictive scheduling, disruption, learning, online scheduling.*

4.1 Intoduction

Earth Observation Satellites (EOS) play an important role in surveillance, tracking or monitoring situations for several real-world domains. The recent generation of Agile EOS (AEOS) has emerged to be highly effective due to their attitude manoeuvring capabilities along three degrees of freedom – roll, pitch and yaw. In the past couple of decades, AEOS scheduling problem has been of much interest to researchers due to its complex search space and vast-ranged applicability requirements (Wang et al.; 2020).

The AEOS scheduling problem comprises finding an optimal assignment of user-requested imaging tasks to respective AEOSs by satisfying the resource constraints in a specified time frame (Hall and Magazine; 1994). The Areas of Interest (AOIs) for these imaging tasks have diverse specifications involving geometry and location of AOIs and time-bounded task completion requirements. These tasks may stem from imaging requirements in the identification, tracking, and surveillance of targets. The workflow of EOS systems generally consists of collecting imaging task requests from the users, planning the optimal task schedule of the EOSs, up-linking the schedule to the EOSs, down-linking the obtained image data from the EOSs and processing the information before sending it to the users. Some of the operational resource constraints in the AEOS scheduling problem involve consideration of the energy requirements due to appropriate attitude manoeuvring for imaging the AOI and having enough memory for data collection (Chatterjee and Tharmarasa; 2022).

The user-requested task schedule is uplinked to the AEOSs during their communications with the ground stations. In a mission, these communications can occur several times. A considerable amount of literature has investigated the optimality of task assignments prior to the start of the mission in the form of an offline optimization framework (Globus et al.; 2003; Habet et al.; 2010; Tangpattanakul et al.; 2015; Liu et al.; 2017; Peng et al.; 2020). In this case, the scheduling is generally performed in a single step before the mission starts. As a result, though the optimization problem is complex in nature due to operational constraints, it does not consider schedule updations for failed tasks due to sensor malfunctions or cloud coverage.

Based on the applications of the imaging tasks, schedule updations during the mission may be required. These can occur due to requirements for the completion of a new task or several new task requests during the mission. Schedule updates may also be required in dynamic scenarios such as the presence of cloud coverage or sensor malfunctions creating uncertainty in task initiation and completion (He et al.; 2019; Wang et al.; 2019). Recent literature on AEOS schedule updations considers online rescheduling when real-time information is available (Chu et al.; 2017; Haijiao et al.; 2019). However, in dynamic environments, complete rescheduling with real-time information requires a huge computational load and valuable time(Zhai et al.; 2015).

This work focuses on dealing with the challenging problem of online scheduling when new tasks arrive during an ongoing mission. The AEOS scheduling problem is traditionally considered difficult to address for its NP-hard nature (Lemaître et al.; 2002). Additionally, the aspect of online real-time scheduling with new tasks arriving during the mission increases the complexity of the problem scenario manifold.

In response to real-time task requests, a completely online scheduling strategy using deep reinforcement learning is employed to maximize expected profit for performing user-requested tasks(Haijiao et al.; 2019). This method requires sequential training in several steps during the mission. Another work on sequential task scheduling using long short-term memory (LSTM) (Peng et al.; 2018) considers tasks known prior to the mission, however, the decision-making is performed in a completely online strategy and does not consider the arrival of new tasks during the mission. However, these completely online scheduling methods may be used for the dynamic arrival of tasks, but are not adequate for daily imaging tasks, an important requirement in real-world AEOS applications. A combination of both offline and online scheduling strategies is necessary to achieve global optimality for the tasks known prior to the mission, and near-optimality for emergent tasks or dynamic environments during the mission. A recent work on the combination of offline and online scheduling for cloud coverage considers an offline pre-assignment of AEOSs to the tasks with a rough schedule without an exact start time for the tasks. Then consequently, an online schedule is proposed to refine the start time using real-time cloud information (He et al.; 2019). However, this above-mentioned work considers only low-orbit EOSs and does not consider new task arrivals. In contrast, the focus of the present work is on dealing with new task arrivals by considering an offline scheduling approach for pre-mission known tasks and an online scheduling approach for new task arrivals.

AEOS scheduling problems have structural similarities with the combinatorial job scheduling problem (Lemaître et al.; 2002). In the context of rescheduling for new job arrivals in a job scheduling problem, a detailed theoretical analysis has been carried out to minimize the change in the original schedule (Hall and Potts; 2004; Hall et al.; 2007). In order to reduce the havoc in original resource allocations, disruption costs to the original schedule are considered an important metric to minimize change to the existing schedule. A similar concept has been adopted in the present work by using a multi-objective optimization framework that considers the trade-off between reward maximization by existing and newly arrived task completions and minimizing disruption to the original schedule for the new tasks.

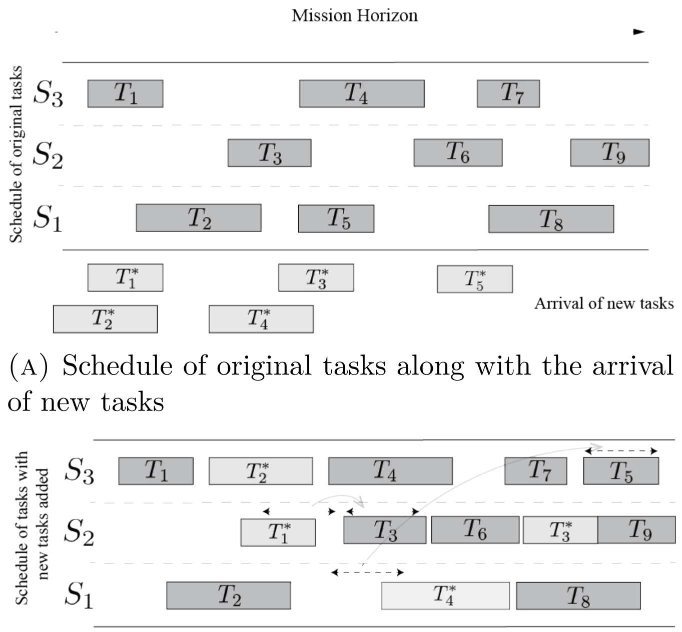
In recent literature on AEOS scheduling problems, machine learning has been used to improve solution quality in online and offline scheduling (Wang et al.; 2020). In the case of online scheduling for new task arrivals, a classifier based on random forests has been implemented to identify whether to accept a new task to integrate into the schedule or not (Lu et al.; 2020). This classifier is further attached to an existing onboard scheduling algorithm (Liu et al.; 2017). The current work is a combination of online and offline strategies of scheduling to address both known tasks before the mission and new task arrivals, a learning-based dummy task estimator is considered on the basis of historical data. These dummy tasks are further used to generate a predictive schedule that minimizes disruption to the original schedule and maximizes the total reward from task completion. The creation of this predictive schedule and ultimately using an assignment problem to match actual new tasks to the dummy tasks is a major contribution of this work.

In section 4.2, the problem statement is formalized. The mathematical formulation of the problem is discussed in section 4.3. Sections 4.4 and 4.5 illustrate the solution methodology and experimental validations using simulations for the proposed predictive scheduling of new tasks. The final section 4.6 concludes the work and discusses the future scopes of this work.

4.2 Problem Statement

The user-specified tasks in an AEOS scheduling problem involve imaging particular geographical locations (AOIs) with multiple AEOSs on their respective orbits. Successful completion of the tasks yields rewards. A single successful scan of the specified AOI would yield the corresponding maximum reward for a task. When the user requests a scanning task during the mission, adequate operational resources and time have to be assigned to ensure the successful accommodation of the task. The satellites can perform these tasks within a certain visible time window (VTW) in their trajectory based on the location of the tasks and the attitude maneuvering capability of the AEOSs in terms of roll and pitch angles. The satellites need to reserve enough energy resources to obtain the appropriate maneuver required in imaging a particular AOI.

The ground stations are distributed across the Earth and are connected as a centralized network. The satellites can communicate with these ground stations only when they are within the VTW of the ground stations. The information obtained from the scanning tasks is down-linked and processed in these ground stations. After the down-link, the onboard memory of the satellites is reset to enable information acquisition from the consequent imaging tasks. The satellites have a limited onboard memory to store the information collected during the scans.



(A) Schedule of original tasks along with the arrival of new tasks

(B) Existing schedule updation to accommodate new tasks

FIGURE 4.1: Updation in existing schedule to accommodate new task arrivals during mission

The main focus of this work lies in the accommodation of newly arrived tasks during an ongoing AEOS mission. The arrival of new tasks may occur multiple times as a unit task or in batches throughout the mission horizon. An unsuccessful scan of a task during the mission can also be considered as a new task in the succeeding time periods of the mission horizon. The existing schedule consisting of the tasks that arrived prior to the start of the mission needs to be updated after each arrival. These updated schedules have to be up-linked from the ground stations to the AEOSs during the communications.

The updation of the existing schedule may be handled through complete rescheduling of all the tasks that were known prior to the mission and the new tasks that have arrived during the mission. This can lead to a high degree of deviation from the existing schedule, which will result in havoc in the form of schedule disruptions, delays in existing task completion, and skipping of existing tasks. The accommodation of new tasks in real-time with a complete reschedule can also result in high computational costs. This work addresses the issues of disruption and high computational cost with a four-phase scheduling strategy. Fig. 4.1 shows an example scenario of the problem considered in this paper. Subfigure 4.1a shows the original schedule of nine tasks assigned to three satellites. It also shows the five new tasks that arrived during the mission. Subfigure 4.1b shows that the original mission schedule has been updated, where some of the tasks have been moved to accommodate the new tasks.

In this work, the consideration of historical data and machine learning-based predictive scheduling is adopted in the pre-mission stages in order to gain insights regarding future task arrivals. This predictive modelling of task arrivals will provide a reflection of the past trends from historical missions and current trends of task arrivals originating during the mission.

4.3 Problem Formulation

Let N_s satellites in the mission have pre-defined trajectories and periodic communications with N_g ground stations. Prior to the start of the mission, a set of J_0 user-requested tasks is to be assigned to the satellites. Let the assignment obtained at this stage for the J_0 tasks be π^* for the entire mission horizon $[0, \mathfrak{T}]$. This assignment will be obtained by maximizing the reward obtained for successfully completing tasks with the optimal roll and pitch angle combinations and satisfaction of resource constraints as described in subsection 4.3.2.

The main focus of this work lies in accommodating the sets of $J_{new}^{\tau_i}$ newly arrived tasks in the existing schedule π^*, where the tasks arrive during the k time periods denoted by $\tau_i \subset [0, \mathfrak{T}], \forall i \in [1, k]$. The disruption to the existing schedule π^* for the insertion of these new $J_{new}^{\tau_i}$ tasks should be minimized to avert delay in task completion and to discard the possibilities of reduction in the potential reward from the existing schedule π^*. Updations in the existing schedule π^* with all the new task arrivals would result in an updated schedule π_{new}^* that minimally disrupts the existing schedule. Since this process of arrival of new tasks is highly dynamic in nature, the optimal generation of an online schedule π_{new}^* could be extremely challenging. The new task arrivals $J_{new}^{\tau_i}, \forall i$ are assumed to be correlated with historical data of task arrivals, hence an approach of learning-based predictive rescheduling, using predicted dummy tasks $\tilde{J}_{new}^{\tau_i}, \forall i$ from historical data, followed by the assignment of actual tasks to the dummy tasks is proposed.

The predictive rescheduling is run prior to the mission to reduce the disruption and complexity and to increase the efficiency of online task scheduling during the mission. The rate of task arrivals is learned with a function approximation, and a set of dummy tasks $\tilde{J}^{\tau_i}_{new}$ is generated on that basis. These $\tilde{J}^{\tau_i}_{new}$ tasks are inserted within the existing schedule π^* with consideration of the resource constraints and the satellite trajectories to create a predictive schedule π_{new} using a multi-objective optimization that minimizes disruption to the existing schedule and maximizes the reward for performing the tasks. Here, π_{new} generates predictive slots of idle times for the satellites using the $\tilde{J}^{\tau_i}_{new}$ tasks. During these idle times, the satellites are available to take up new tasks if needed during the mission. Following this predictive reschedule, π_{new}, an assignment optimization problem is solved during the mission for appropriate matching of newly arrived actual tasks, $J^{\tau_i}_{new}$ to the dummy tasks, $\tilde{J}^{\tau_i}_{new}$. If some new tasks cannot be matched with the dummy tasks, they have to be assigned within the schedule containing tasks known prior to the mission and the new tasks already matched with the dummy tasks. The predictive rescheduling thus ensures that the computational requirements of scheduling only these unmatched new tasks during the mission are much lesser than rescheduling for all the new task arrivals during the mission.

The complete optimization model is illustrated in Fig. 4.2. The first three phases run prior to the mission, and phase four runs during the mission, in a combination of offline-online setup. The four phases of the optimization model are described in a detailed manner in the following subsections.

4.3.1 Phase 1 – Learning the historical trend of new task arrival

(Prior Mission)

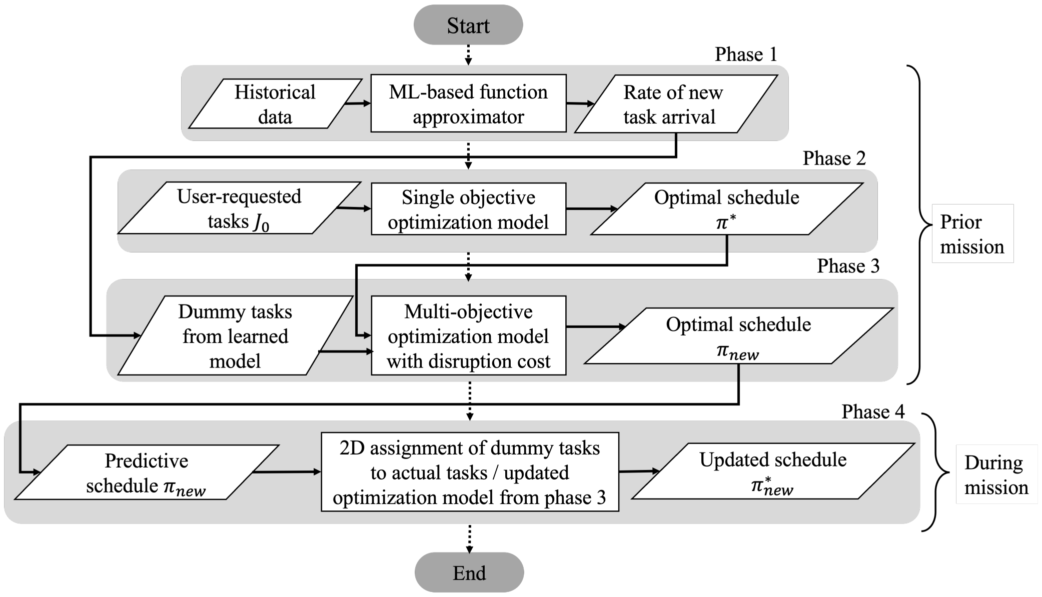

Start
Phase 1
Historical data
ML-based function approximator
Rate of new task arrival
Phase 2
User-requested tasks J_0
Single objective optimization model
Optimal schedule π^*
Phase 3
Dummy tasks from learned model
Multi-objective optimization model with disruption cost
Optimal schedule π_{new}
Prior mission
Phase 4
Predictive schedule π_{new}
2D assignment of dummy tasks to actual tasks / updated optimization model from phase 3
Updated schedule π^*_{new}
During mission
End

Figure 4.2: Multi-stage optimization model with learning-based predictive scheduling for new task arrivals

It is assumed that the time horizon is partitioned into k intervals, as shown in Fig. 4.3. In order to learn the pattern of potential new task arrivals, a function approximator is used that generates the predicted new task arrival rate at each time interval. The training data for this predictive function approximator would be the historical data of new task arrivals for the past $\mathcal{N}$ days. Let the variable $\left|J_m^{\tau_i}\right|, m \in [1, \mathcal{N}], i \in [1, k]$ denote the historical number of task arrivals on day m at the time period τ_i.

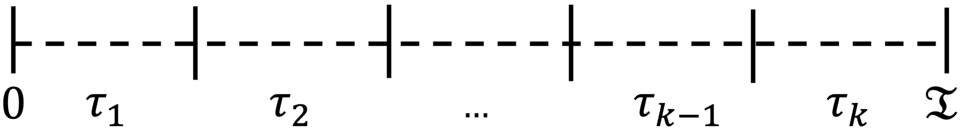


Figure 4.3: Mission horizon

There may exist correlations in the rate of task arrivals in consecutive time periods. In order to capture any hidden pattern between preceding time periods and the current time period, this function approximator treats this information of task arrivals as the input and learns the hidden pattern, if it exists. Let the input consist of the number of task arrivals for the preceding κ time periods and the index of the current time period, i. Then the input for the function approximator may be defined as the tuple given by $\left\langle \left|J_m^{\tau_{i-1}}\right|, \left|J_m^{\tau_{i-2}}\right|, \ldots, \left|J_m^{\tau_{i-\kappa}}\right|, i \right\rangle$. The output is defined as $\left|\tilde{J}_m^{\tau_i}\right|$, which is the predicted number of task arrivals in the time period i. The correlation of the rate of task arrivals in consecutive time periods is learned by training the function approximator on all of the $i \in [\kappa + 1, k]$ time periods. This means that if the trained function approximator knows which time interval it is predicting and the number of task arrivals for the preceding time periods, it is able to predict the number of task arrivals in any time period throughout the mission horizon.

The arrival of new tasks might vary from one region to another and may have correlations within the neighboring regions. In the case of surveillance and tracking tasks, the target may move from one region to the neighboring region. As a result, the number of required imaging tasks to fulfill this surveillance will have a correlation within these neighboring regions. Learning may be used to detect the pattern of these correlations, in the scenarios, where this pattern is maintained over historical data.

Let the total mission region be split into equal-sized L sub-regions. Depending upon the tasks in the historical data, the size of each sub-region may be defined. Let the variable $\left|J_m^{R_l}\right|, m \in [1, \mathcal{N}], l \in [1, L]$ denote the number of task arrivals on day m in region R_l. Combining the rate of task arrivals corresponding to respective time intervals and regions from the historical data, an updated variable $|J_m^{\tau_i, R_l}|, m \in [1, n], i \in [1, k], l \in [1, L]$ is considered. In order to learn the hidden pattern of the rate of task arrivals, the inputs of the function approximator may be defined by the tuple $\Big\langle |J_m^{\tau_{i-1}, R_{l_1}}|, |J_m^{\tau_{i-2}, R_{l_1}}|, \ldots, |J_m^{\tau_{i-\kappa}, R_{l_1}}|, |J_m^{\tau_{i-1}, R_{l_2}}|, |J_m^{\tau_{i-2}, R_{l_2}}|, \ldots, |J_m^{\tau_{i-\kappa}, R_{l_p}}|, |J_m^{\tau_{i-1}, R_l}|, |J_m^{\tau_{i-2}, R_l}|, \ldots, |J_m^{\tau_{i-\kappa}, R_l}|, \tau_i, R_l \Big\rangle$, where $R_{l_1}, R_{l_2}, \ldots, R_{l_p}$ are the adjacent p-point neighbors of the region R_l. The output of the function approximator provides a predicted rate of arrival $\left|\tilde{J}_m^{\tau_k, R_l}\right| \; \forall i \in [1, K], l \in [1, L]$. If there is no correlation between the region and its neighbours, the information of the neighbours can be omitted from the input tuple.

The training of the function approximator with the historical data is performed by minimizing the difference between the predicted value and the actual value. This error calculation is done on the basis of the mean squared error, which can be calculated as follows,

$$MSE\left(\epsilon^{\tau_k, R_l}\right) = \frac{1}{n} \sum_{n} \left| \left|\tilde{J}_m^{\tau_i, R_l}\right| - \left|J_m^{\tau_i, R_l}\right| \right|^2 \tag{4.1}$$

The parameters of the function approximator are tuned in each of the training epochs while minimizing the Mean Squared Error (MSE) at each epoch. An ML-based solution technique is discussed in section 4.4.1.

4.3.2 Phase 2: Assignment of J_0 tasks

(Prior Mission)

The learning-based prediction of the rate of task arrivals provides an estimate of the number of tasks that may arrive during the mission horizon. However, since the set of J_0 tasks arrive prior to the mission initiation, the assignment of these tasks π^* needs to be performed prior to the mission. In this subsection, the phase of the optimization strategy, where J_0 tasks are assigned, is illustrated (Chatterjee and Tharmarasa; 2022).

The objective function for the assignment problem involves maximizing the reward accumulated by all the satellites during the mission by performing the successful scan of the J_0 tasks, which may be defined as follows,

$$\max_{a_{(s,o,t)},\ T^{st}_{(s,o,t)}} \sum_{t=1}^{|J_0|} \sum_{s=1}^{N^s} \sum_{o=1}^{N^{os}} a_{(s,o,t)} R_{(s,\alpha,t)} \tag{4.2}$$

where t, s, o indicate the indices for J_0 tasks, N^s satellites, and the corresponding N^{os} revolutions, respectively. Here, $a_{(s,o,t)}$ is a binary decision variable, which denotes whether task t is assigned to s in revolution o ($a_{(s,o,t)} = 1$) or not ($a_{(s,o,t)} = 0$) and $T^{st}_{(s,o,t)}$ denotes the start time of the scan. The reward $R_{(s,\alpha,t)}$ achieved by scanning depends on the observation angle α of the task t from the satellite s defined by the start time of the task $T^{st}_{(s,o,t)}$. The correlation between the reward and observation angle is dependent on the task and is determined by the user. Typically, the reward decreases as the observation angle increases. To calculate the angle α, the satellite's roll and pitch angle movements are combined based on the relative location of the task and the satellite illustrated in Fig. 4.4. For example, the relationship of the observation angle with the reward may be calculated as, (Chatterjee and Tharmarasa; 2022)

$$R_{(s,\alpha,t)} = R_t^{max} - \left(\frac{R_t^{max}}{2} \times \frac{\theta + \phi}{\theta_{max} + \phi_{max}} \right) \tag{4.3}$$

where R_t^{max} is the maximum possible reward to accumulate for task t. Here, θ and ϕ are the roll and pitch angles for the scan, and θ_{max} and ϕ_{max} are the maximum allowable roll and pitch angles for the satellite.

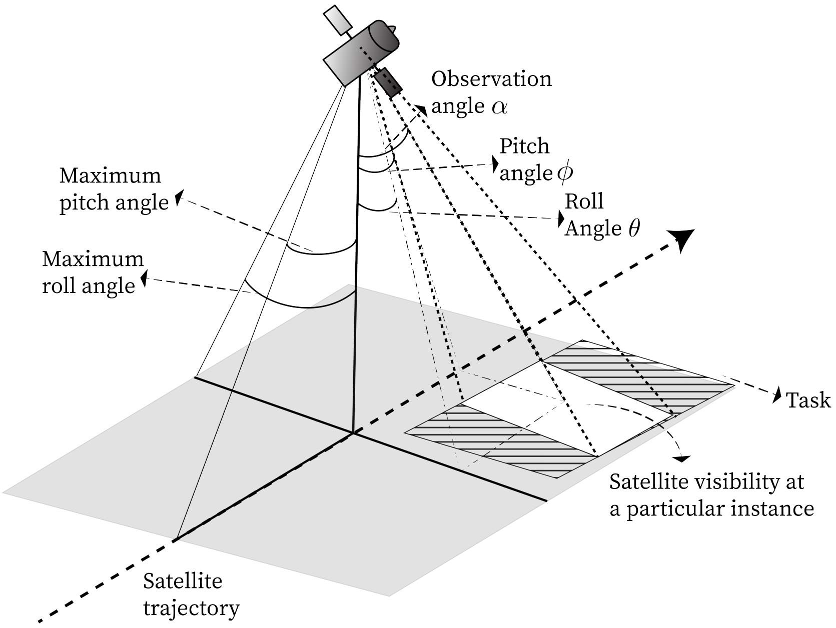


Figure 4.4: Observation angle of a satellite

The objective function maximizing the reward accumulated for J_0 tasks is subject to several resource constraints originating from satellite specifications and trajectories. These constraints are defined as follows.

Let the available charge at the starting time for a scan of task t by satellite s be denoted as $C_{\left(s,T^{st}_{(s,o,t)}\right)}$. The energy $e_{(s,o,t)}$ required to scan task t in revolution o should always be lesser than or equal to the available charge at the starting time $C_{\left(s,T^{st}_{(s,o,t)}\right)}$ of the scan and the gained energy $ch_s \times d_{(s,o,t)}$ during the scan, where ch_s is the charging rate of the satellite and $d_{(s,o,t)}$ is the duration of scanning the task t by satellite s in revolution o. The energy constraint can be defined as

$$\left[C_{\left(s,T^{st}_{(s,o,t)}\right)} + ch_s d_{(s,o,t)} - e_{(s,t)}\right] \geq \rho \qquad \forall s, o, t \text{ when } a_{(s,o,t)} = 1 \tag{4.4}$$

where ρ is the minimum energy level the satellite needs to maintain. The value of ρ can be zero if the energy level of the satellites can be as minimum as zero.

It is important to ensure that the satellite has enough energy to rotate. The total available energy at the start of the rotation is given by $C_{\left(s,T^{rot}_{(s,o,t,\hat{t})}\right)}$, where $T^{rot}_{(s,o,t,\hat{t})}$ is the start time of the rotation. The energy gained by the satellite during the roll and pitch movements is given by $ch_s\left(d^{roll}_s \theta^{t,\hat{t}}_{(s,o)}\right)$ and $ch_s\left(d^{pitch}_s \phi^{t,\hat{t}}_{(s,o)}\right)$, respectively, where $\theta^{t,\hat{t}}$ and $\phi^{t,\hat{t}}$ are the required rotations along the roll and pitch angles, respectively, while transitioning from task t to task $\hat{t}$, and d^{roll}_s and d^{pitch}_s are the duration for each roll and pitch angle.

$$C_{\left(s,T^{rot}_{(s,o,t,\hat{t})}\right)} + ch_s\left(d^{roll}_s \theta^{t,\hat{t}}_{(s,o)} + d^{pitch}_s \phi^{t,\hat{t}}_{(s,o)}\right) - \left(\theta^{t,\hat{t}}_{(s,o)} e^{roll}(s) + \phi^{t,\hat{t}}_{(s,o)} e^{pitch}(s)\right) \geq \rho$$
$$\forall s, o, t \in [1, |J_0|], \hat{t} \neq t \tag{4.5}$$

where $e^{roll}(s)$ and $e^{pitch}(s)$ are the energy needed for each roll and pitch angle change, respectively.

The memory constraints are illustrated in equations (4.6) and (4.8). While scanning, the on-board memory of the satellite is utilized, and the memory is reset during the communication with the ground station. Let $m_{(s,t)}$ be the memory used by satellite s for task t and m^{max}_s be the maximum capacity of the satellite s. The satellite s can only perform a specific number of scans in between the communication to ground station g and the next available ground station $\hat{g}$. The value of $N_{(s,g,\hat{g})}$ is calculated from the decision variable $T^{st}_{(s,o,t)}$, which denotes the start time of the scan in the interval between two consecutive communications with the ground station g and $\hat{g}$ for AEOS s.

$$\sum_{N_{(s,g,\hat{g})}} m_{(s,t)} \leq m^{max}_{s} \qquad \forall s, g, \hat{g} \neq g \tag{4.6}$$

The memory constraints (4.7) and (4.8) ensure that the communication between the ground station and the satellites does not occur during a scan, where d^{mem}_{s} is the time needed for satellite s to communicate with ground stations.

$$T^{st}_{(s,o,t)} > d^{mem}_{s} + T^{sg}_{(s,o,g)} \qquad \forall s, o, t, g \tag{4.7}$$

$$T^{sg}_{(s,o,g)} > T^{st}_{(s,o,t)} + d_{(s,o,t)} \qquad \forall s, o, t, g \tag{4.8}$$

The timing constraints (4.9) and (4.10) ensure that the start time of communication between the ground station and the satellite, as well as the start time of the tasks, are within the VTW of the satellites. The VTW of the ground station g is defined as $\left[T^{VTW_{st}}_{(s,o,g)}, T^{VTW_{end}}_{(s,o,g)}\right]$ and the VTW for task t is defined as $\left[T^{VTW_{st}}_{(s,o,t)}, T^{VTW_{end}}_{(s,o,t)}\right]$.

$$T^{VTW_{st}}_{(s,o,g)} \leq T^{st}_{(s,o,g)} \leq T^{VTW_{end}}_{(s,o,g)} - d^{mem}_{s} \qquad \forall s, o, t \tag{4.9}$$

$$T^{VTW_{st}}_{(s,o,t)} \leq T^{st}_{(s,o,t)} \leq T^{VTW_{end}}_{(s,o,t)} - d_{(s,o,t)} \qquad \forall s, o, t \tag{4.10}$$

The constraint (4.11) checks if there is enough time between the start time of the next task and the end time of the previous task for the satellites to manoeuvre the sensor to the appropriate roll and pitch angles.

$$T^{st}_{\left(s,o,\hat{t}\right)} \geq T^{st}_{(s,o,t)} + d_{(s,o,t)} + d_s^{roll}\theta^{t,\hat{t}}_{(s,o)} + d_s^{pitch}\phi^{t,\hat{t}}_{(s,o)} \qquad \forall s,o,t \tag{4.11}$$

In this work, it is assumed that the start time of the communications, $T^{st}_{(s,o,g)}$, is pre-determined. However, it may be considered as a decision variable as done in (Chatterjee and Tharmarasa; 2022).

4.3.3 Phase 3: Predictive scheduling for new task arrivals (Prior Mission)

In the previous phase, J_0 tasks are assigned to the satellites at the start of the mission. In order to obtain the dummy tasks $\tilde{J}^{\tau_i}_{new}$ for time period $\tau_i, \forall i \in [1,k]$, the output of the trained function approximator from phase one is used. The number of tasks (including the existing and the new dummy tasks) to be assigned to the satellites in consecutive time intervals can be defined as follows,

$$|\mathcal{J}^{\tau_i}| = \begin{cases} |J_0| & \text{at time } 0 \\ |\hat{J}^{\tau_1}|(1-\xi_1) + |\tilde{J}^{\tau_1}_{new}| & \text{at time } \tau_1 \\ |\hat{J}^{\tau_2}|(1-\xi_2) + |\tilde{J}^{\tau_2}_{new}| & \text{at time } \tau_1+\tau_2 \\ \cdots & \cdots \\ |\hat{J}^{\tau_k}|(1-\xi_k) + |\tilde{J}^{\tau_k}_{new}| & \text{at time } \tau_1+\tau_2+\cdots+\tau_k \end{cases} \quad , \tau_i \subset [0, \mathfrak{T}] \qquad (4.12)$$

where $|\hat{J}^{\tau_i}(\subset J_0)|$ is the number of tasks scheduled for the time interval $\tau_i,\ i \in [1,k]$ in phase 2. Here, ξ_i is the probability of a successful scan for the time interval τ_i. The unsuccessful scans of the existing tasks J_0 at the start of the time period τ_i are denoted by $\hat{J}^{\tau_i}(1-\xi_i)$.

At this stage, a predictive reschedule is performed on the original task requests J_0 and the newly arrived dummy tasks to avoid creating havoc and to reduce the scheduling costs. This predictive schedule π_{new} has to be generated in a manner that produces the least disruption to the existing schedule π^*. The disruption cost measures the difference in the start times of the existing J_0 tasks in π^* and π_{new}. A metric for disruption cost in terms of completion times of a job in job scheduling problems can be referred to in (Hall et al.; 2007). Either way, minimizing the disruption ensures the minimal change in the tasks for the existing schedule π^*.

The disruption cost for task t due to the updated schedule π_{new} can be defined by,

$$\Delta_t(\pi^*, \pi_{new}) = \left| T^{st}_{(s,o,t)}(\pi^*) - T^{st}_{(s',o',t)}(\pi_{new}) \right| \quad \forall t \in J_0 \qquad (4.13)$$

Since there exists a trade-off between minimizing disruption in the existing schedule and maximizing the potential reward in the updated schedule with the newly arrived tasks, the following multi-objective optimization problem is addressed at this stage.

$$f_1 = \underset{a_{(s,o,t)},\ T^{st}_{(s,o,t)}}{\arg} \max \sum_{t=1}^{|J|} \sum_{s=1}^{N^s} \sum_{o=1}^{N^{os}} a_{(s,o,t)} R_{(s,\alpha,t)}, \quad \text{where } J = \sum \mathcal{J}^{\tau_i},\ \tau_i \subset [0, \mathfrak{T}] \tag{4.14}$$

$$f_2 = \underset{T^{st}_{(s,o,t)}}{\arg} \min \sum_{t=1}^{|J_0|} \left| T^{st}_{(s,o,t)}(\pi^*) - T^{st}_{(s',o',t)} \right| \tag{4.15}$$

subject to the constraints defined in equations (4.4 – 4.11). The potential reward to be achieved at this stage $R_{(s,\alpha,t)}$ includes the potential completion of the predicted dummy tasks, failed tasks and the tasks from J_0. The individual rewards for the predicted dummy tasks are considered to be much lower than the reward achievement for J_0 tasks to ensure the optimization problem does not skip the original tasks. The skipped tasks are also selected randomly based on the failure probability due to the weather condition of a particular location from the set $\hat{J}^{\tau_i}$, and the reward is set to a lower value as well due to the uncertainty of the failure.

Since the newly arrived dummy tasks have to be scheduled after their arrival, an additional constraint has to be considered, given as,

$$\tau_i < T^{st}_{(s,o,t^*)} < \mathfrak{T}, \quad \forall t^* \in \tilde{J}^{\tau_i}_{new} \tag{4.16}$$

4.3.4 Phase 4: Onboard scheduling in real-time (During Mission)

Based on the number of actual new task arrivals, $|J^{\tau_i}_{new}|$ compared to the number of predicted dummy tasks, $|\tilde{J}^{\tau_i}_{new}|$, along with the predicted missed tasks $|\hat{J}^{\tau_i}|(1-\xi_i)$ and actually missed tasks $|J^{\tau_i}_{miss}|$ the following two scenarios may arise,

- *Case 1:* $|J^{\tau_i}_{act}| \leq |J^{\tau_i}_{pred}|$
- *Case 2:* $|J^{\tau_i}_{act}| > |J^{\tau_i}_{pred}|$

where $|J^{\tau_i}_{act}| = |J^{\tau_i}_{new}| + |J^{\tau_i}_{miss}|$ and $|J^{\tau_i}_{pred}| = |\tilde{J}^{\tau_i}_{new}| + |\hat{J}^{\tau_i}|(1-\xi_i)$. For cases 1 and 2, the ground station scheduler accommodates the new tasks $|J^{\tau_i}_{act}|$ in the idle spaces due to the dummy tasks $|J^{\tau_i}_{pred}|$ with the assignment problem defined below. The scheduler then tries to accommodate the rest of the tasks to the pre-existing idle spaces in the schedule other than the idle spaces created from the dummy tasks. Some of the tasks which are still not assigned are kept for rescheduling. The probability of rescheduling is higher for case 2 than case 1 due to the excess number of actual new tasks.

4.3.4.1 Assignment of actual tasks to predicted tasks

The assignment of the new tasks to idle spaces is done by matching the predicted tasks of $J^{\tau_i}_{pred}$ and actual tasks of $J^{\tau_i}_{act}$. The method of matching these tasks is essentially a linear assignment problem in combinatorial optimization.

Let a binary assignment variable $\mathcal{A}_i(p, q)$ that associates the tasks for the time interval $\tau_i,\ i \in [1, k]$ be defined as,

$$\mathcal{A}_i(p,q) = \begin{cases} 1 & \text{Task } J^{\tau_i}_{pred}(p) \text{ is associated to task } J^{\tau_i}_{act}(q) \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

Here, the association indices for $J^{\tau_i}_{pred}$ and $J^{\tau_i}_{act}$ are given by p and q, respectively. Then, the set of assignments for all the associations for tasks in $J^{\tau_i}_{act}$ to $J^{\tau_i}_{pred}$ for the time interval τ_i can be defined as,

$$\mathbf{A}_i = \{\mathcal{A}_i(p,q);\ p = 0, 1, \ldots, |J^{\tau_i}_{pred}|;\ q = 0, 1, \ldots, J^{\tau_i}_{act}\} \tag{4.18}$$

Let the indices $p = 0$ and $q = 0$ be the indices for non-existent tasks. Assignment $\mathcal{A}_i(0, q)$ denotes that actual task $J^{\tau_i}_{pred}(q)$ is not associated to any dummy task in $J^{\tau_i}_{act}$. Equivalently, Assignment $\mathcal{A}_\rangle(p, 0)$ signifies that dummy task $J^{\tau_i}_{act}(q)$ is not associated with any actual task in $J^{\tau_i}_{pred}$.

This assignment is performed by a single-objective optimization problem that minimizes the cost of the association of the actual tasks to the dummy tasks. The objective of the assignment is to find the optimal assignment $\mathbf{A}^*{}_i$ given by,

$$\mathbf{A}^*_i = \arg\max_{\mathbf{A}_i} \sum_{p=0}^{|J^{\tau_i}_{pred}|} \sum_{q=0}^{|J^{\tau_i}_{act}|} \mathcal{A}_i(p,q)\ c_i(p,q) \tag{4.19}$$

where $c_i(p, q)$ is the reward for the assignment $\mathcal{A}_i(p, q)$. The value of the $c_i(p, q)$ is the weighted sum of reward for task q and inversely proportional distance between tasks p and q. For example, in the simulation of this paper, the following equation has been used.

$$c_i(p,q) = w_1 R_q^{max} + w_2 \frac{1}{dist(p,q)} \tag{4.20}$$

where $dist(p,q)$ is the locational distance between task p and q. The inversely proportional reward for the distance ensures that the distances between the dummy tasks and the actual tasks are minimized to reduce the effect of roll and pitch angle in the reward accumulated and the energy constraint. The values of w_1 and w_2 can be tuned based on the applications of the task.

This assignment problem is optimized by satisfying the constraints given below,

- *One-to-one mapping constraints:*

$$\mathcal{A}_i(p,q) \wedge \mathcal{A}_i(p',q) \neq 1 \quad \forall p \neq p';\; p,p' \neq 0 \tag{4.21}$$

$$\mathcal{A}_i(p,q) \wedge \mathcal{A}_i(p,q') \neq 1 \quad \forall q \neq q';\; q,q' \neq 0 \tag{4.22}$$

These constraints ensure that exactly one actual task is associated with exactly one dummy task. The only exception occurs for the non-existent tasks ($p = 0, q = 0$), which can be associated with multiple actual tasks and multiple dummy tasks.

- *Start time constraint:*

$$T^{st}_{(s,o,q)} = T^{st}_{(s,o,p)} \pm \delta_q^{T_{st}} \tag{4.23}$$

The start time of task q should also be closer to the start time of task p. The term $\delta_q^{T_{st}}$ ensures that even though the start time is not exactly the same, the actual task q can still be assigned to the dummy task p when there is enough idle time. This can be defined by the following equations (4.24) and

(4.25) in the context of tasks t and t' that are already scheduled.

$$T^{st}_{(s,o,q)} + d_{(s,o,q)} < T^{st}_{(s,o,t)}, \quad \text{where task } t \text{ is followed by task } q \tag{4.24}$$

$$T^{st}_{(s,o,t')} + d_{(s,o,t')} < T^{st}_{(s,o,q)}, \quad \text{where task } q \text{ is followed by task } t' \tag{4.25}$$

In order to minimize the computational complexity during the mission, the value of $\delta^{T_{st}}_{q}$ is chosen as small as possible to set the start time of task q as close to task p as possible when it follows the equations (4.24) and (4.25).

- *Roll and pitch angle constraints:*

$$\theta(q) \leq \theta_{max} \tag{4.26}$$

$$\phi(q) \leq \phi_{max} \tag{4.27}$$

These constraints ensure that during the mapping, the roll and pitch angles assigned to the actual task (i.e., $\theta(q), \phi(q)$) are within the maximum limit of the roll and pitch angles.

4.3.4.2 Rescheduling of the remaining tasks

The remaining tasks are attempted to be assigned to the limited idle spaces available in the existing schedule. A modified optimization model can be used similar to the proposed phase 2, with only the remaining tasks and an additional constraint that only the idle spaces are available.

$$[T^{st}_{(s,o,t)}, T^{st}_{(s,o,t)} + d_{(s,o,t)}] \in \mathcal{D}(\pi^{*}_{new}) \tag{4.28}$$

where $\mathcal{D}(\pi^*_{new})$ is a vector variable for the idle time slots for the schedule π^*_{new}.

For the scheduling of the remaining tasks, which were not associated with any dummy task or can not be placed in the idle spaces, the multi-objective optimization model from phase 3 is used. Here, the set of initial tasks used in phase 3, J_0 is updated with the tasks that are already scheduled at this point during the mission by the assignment problem. Then the set of initial tasks now becomes $J'_0 = J_0 + J^{\tau'_i}_{new}$, where $J^{\tau'_i}_{new}$ is the set of tasks scheduled during the above-mentioned method of assignment. The dummy tasks in the phase 3 optimization model are now replaced with the remaining tasks $J^{\tau_i}_{new} - J^{\tau'_i}_{new}$ from the assignment problem.

4.4 Solution Methodology

4.4.1 Phase 1:

In recent decades, neural networks are widely used as acceptable function approximators in a range of learning tasks having dynamic and complex information (Scarselli and Tsoi; 1998; Ferrari and Stengel; 2005; Yang et al.; 2013; Sutton and Barto; 2018; Elfwing et al.; 2018).

For predicting the rate of task arrivals using the historical data, a feed-forward fully connected neural network has been used (Svozil et al.; 1997). During the training process, the neural network updates the hyper-parameters by minimizing the MSE as discussed in equation (4.1). This ensures that the hyper-parameters are tuned according to the historical data and can provide high accuracy in the prediction of the task arrival rate. This learning approach in the first phase of the optimization framework generates the estimated rate of new task arrivals during a specific time period in a particular region.

4.4.2 Phase 2:

The goal of the scheduling process in phase 2 lies in finding an optimal schedule from equation (4.2) for the tasks J_0. In (Chatterjee and Tharmarasa; 2022), an elitist mixed-coded genetic algorithm-based satellite scheduling (EMCGA-SS) is proposed that provides efficient assignment in small and large-scale agile multi-satellite scheduling. This algorithm handles uncertainty in task completion by the satellites along with multiple scanning requirements for the tasks. The comparisons of the performance of EMCGA-SS with other meta-heuristics algorithms, such as tabu search (Habet et al.; 2010), and simulated annealing (Wu et al.; 2017) are illustrated in (Chatterjee and Tharmarasa; 2022).

The structure of the chromosome is a hybrid between binary and continuous variables corresponding to the assignment of tasks to the satellites in their specific revolutions and the starting time of the tasks within the VTW, respectively. The processes of crossover and mutation are consistent with the mixed-coded nature of the candidate chromosomes representing the assignment schedules. It is also shown that an initialization with a population of candidate solutions generated with a hill climber enhances the performance of the algorithm EMCGA-SS. A similar approach based on the elitist mixed-coded genetic algorithm may be used to obtain the schedule π^* that is generated from the task set J_0. The required modifications in the algorithm would be the consideration of single scans for each task and a learning-based estimation of task failures. Fig. 4.5 shows the modified chromosome used in this work.

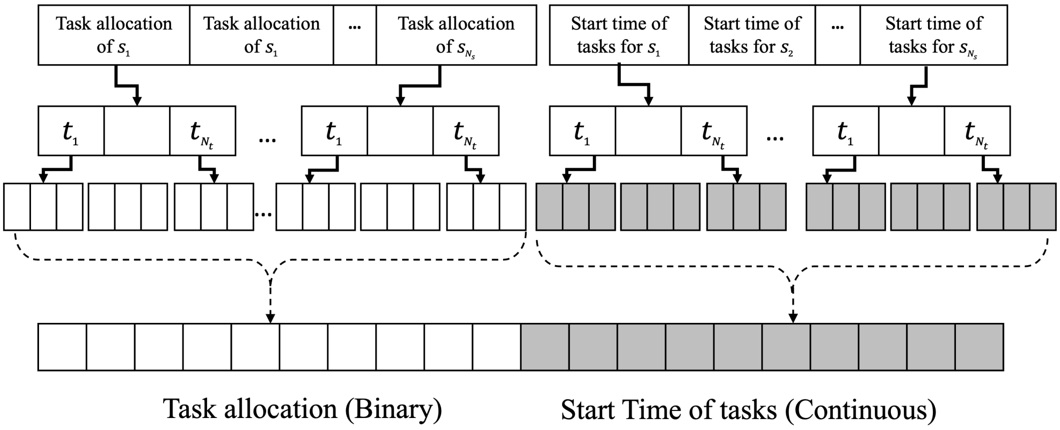

Task allocation of s_1
Task allocation of s_1
...
Task allocation of s_{N_s}
Start time of tasks for s_1
Start time of tasks for s_2
...
Start time of tasks for s_{N_s}
t_1
t_{N_t}
...
Task allocation (Binary)
Start Time of tasks (Continuous)

Figure 4.5: Structure of chromosome

4.4.3 Phase 3:

In multi-objective optimization, dominance relationships between candidate solutions are the key factor when comparing objective function values (Deb; 2011). In the case of this optimization framework, candidate solutions are assignment schedules that illustrate the assignment of satellites to specific tasks in specific revolutions as defined by the trajectories and the corresponding starting times of the tasks. For the multi-objective problem (f_1, f_2), defined in equations (4.14) and (4.15), where f_1 is a maximization objective function and f_2 is a minimization objective function, the dominance between two candidate solutions, i.e., two schedules can be defined as follows,

Definition 1 *The assignment schedule π^x_{new}* ***dominates*** *the assignment schedule π^y_{new}, $(\pi^x_{new} >_{dominates} \pi^y_{new})$, if both of the following conditions are satisfied.*

- *The assignment schedule π^x_{new} is no worse than π^y_{new} in all objectives, given by,*

$$f_1(\pi^x_{new}) \;\geq\; f_1(\pi^y_{new}) \tag{4.29}$$

$$f_2(\pi^x_{new}) \;\leq\; f_2(\pi^y_{new}) \tag{4.30}$$

- *The assignment schedule π^x_{new} is strictly better than π^y_{new} in at least one of the objectives, given by,*

$$\left(f_1\left(\pi^x_{new}\right) > f_1\left(\pi^y_{new}\right)\right) \;\vee\; \left(f_2\left(\pi^x_{new}\right) < f_2\left(\pi^y_{new}\right)\right) \;=\; 1 \tag{4.31}$$

Definition 2 *Two assignment schedules π^x_{new} and π^y_{new} are **non-dominating**, if neither $(\pi^x_{new} >_{dominates} \pi^y_{new})$ nor $(\pi^y_{new} >_{dominates} \pi^x_{new})$ hold, given by,*

$$(f_1(\pi^x_{new}) > f_1(\pi^y_{new})) \;\vee\; (f_2(\pi^x_{new}) < f_2(\pi^y_{new})) \;=\; 0 \tag{4.32}$$

Algorithm 4: Hill-Climber Style Multi-Objective Predictive Scheduler

```
Set initial population $\mathcal{P}$ empty; Initialise feasible schedule $\pi^x_{new}$ from $\pi^*$
  and add to population $\mathcal{P}$;
while termination condition not reached;
 do
    Randomly pick $\pi^y_{new}$ from $P$ and mutate to get $\pi^{y'}_{new}$; Repeat the step if
      $\pi^{y'}_{new}$ is not feasible; If $\pi^{y'}_{new}$ is not dominated by any solution in $\mathcal{P}$ and
      $\pi^{y'}_{new} \notin \mathcal{P}$,
    add $\pi^{y'}_{new}$ to $\mathcal{P}$ and discard all solutions in $\mathcal{P}$ that $\pi^{y'}_{new}$ dominates;
```

The multi-objective optimization approach for generating the predictive schedule with dummy tasks is illustrated in Algorithm 4. The termination condition could be the maximum number of iterations reached.

4.4.3.1 Structure of Chromosome

The structure of the chromosome is an updated structure used during the solution methodology. Fig. 4.6 illustrates an example of how the structure is updated from the original mission schedule chromosome structure. When a new task arrives to the mission, one extra bit is added to each revolution of each satellite in the binary bit section. Consecutively, the same is done in the continuous bit section as well. A random number between 0 and 1 is assigned to the added bit in the binary section. The corresponding bit in the continuous section to a 1 bit in the binary section is assigned a random number in the VTW of the task for the particular satellite in the particular orbit. If the chromosome is not feasible, then the process is repeated. In Fig. 4.6, it is shown that an extra bit is added to each revolution of each satellite. However, a pre-processing of the chromosomes can be implemented to reduce the size of the chromosome. Only the revolutions of those satellites during which it is feasible to perform the scanning tasks are added to the chromosome.

4.4.3.2 Mutation of a schedule

The mutation operation is performed on the chromosomes to move from one schedule to the next. Here, the commonly used mutation rate $1/m$ is considered (Doerr et al.; 2017), where m is the string size of the chromosome. To mutate, on the binary bit, a random bit from the binary bit section is picked, and the bit value is flipped. For the continuous bits representing the start-time of the scan, the bit corresponding to the bit selected in the binary bit is mutated. If the binary bit is flipped from 1 to 0, the corresponding continuous bit is also set to 0. If the binary bit is set to 1 from 0, a random number from the VTW of the task is chosen as the bit value.

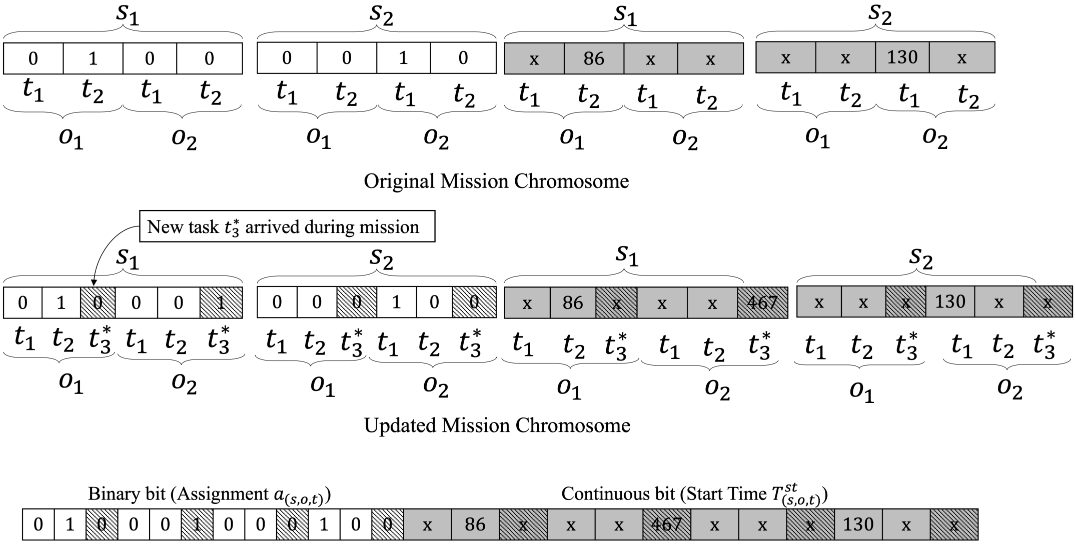

s_1 s_2 s_1 s_2
0 1 0 0 | 0 0 1 0 | x 86 x x | x x 130 x
t_1 t_2 t_1 t_2
o_1 o_2
Original Mission Chromosome
New task t_3^* arrived during mission
0 1 0 0 0 1 | 0 0 0 1 0 0 | x 86 x x x 467 | x x x 130 x x
t_1 t_2 t_3^* t_1 t_2 t_3^*
o_1 o_2
Updated Mission Chromosome
Binary bit (Assignment $a_{(s,o,t)}$)
Continuous bit (Start Time $T^{st}_{(s,o,t)}$)
0 1 0 0 0 1 0 0 0 1 0 0 x 86 x x x 467 x x x 130 x x

FIGURE 4.6: Sample structure of chromosome

4.4.4 Phase 4

Since this phase is performed during the mission, it is important to solve the assignment problem with a fast algorithm. For the linear assignment problem, the Hungarian algorithm (Kuhn; 2010) is widely used. Since this algorithm can solve the assignment problem in polynomial time, it is used for the proposed method.

4.5 Simulations

In this section, the above-mentioned formulation with the proposed solution methodology has been illustrated with experiments on simulated scenarios. The experiments are performed using MATLAB r2020a in an Intel i7, 10th generation processor with 16GB of RAM. The simulated scenarios corresponding to the number of new task arrivals and the proposed solution strategy are described in this section in a detailed manner. Let there be five identical AEOSs in the mission. The trajectories of the AEOSs are predefined. From the trajectory of the AEOSs, it is calculated that one revolution takes around 1 hour 20 minutes for each AEOS. The maximum roll and pitch angles for each AEOS are set to 50° and 60°, respectively. Table 4.1 illustrates the orbital elements for the 5 satellites used in the simulation in the two-line element (TLE) format. The mission horizon for these simulations is set to 12 hours.

TABLE 4.1: Orbital elements of the satellites in TLE format

Sat 1	1 99999U 15182.00000000 .00000125 00000-0 12143-4 0 00008 2 99999 097.7544 188.7205 0002363 359.8062 142.7546 14.92589026000017
Sat 2	1 99999U 14350.00034722 .00000061 00000-0 59555-5 0 00000 2 99999 097.7404 354.5712 0012530 325.4255 209.1276 14.92590498000011
Sat 3	1 99999U 14350.00034722 -.00000135 00000-0 -13100-4 0 00009 2 99999 097.7404 354.5711 0012490 325.3906 089.1576 14.92590756000013
Sat 4	1 99999U 14350.00034722 -.00000125 00000-0 -10109-4 0 00009 2 99999 097.7504 254.5918 0012630 345.4306 109.1476 14.92590156000013
Sat 5	1 99999U 15230.00022326 -.0000075 00000-0 -11105-4 0 00007 2 99999 097.7404 178.3841 0012490 327.3726 136.1735 14.92590756000013

In the first phase, the historical data for the tasks has been simulated for 500 days. The North American region (latitude: $(30°N - 75°N)$, longitude: $60°W - 165°W)$ has been divided into 5×11 grids with each grid of size $10° \times 10°$ latitude and longitude. For each of the 24 time periods, each of duration 30 minutes, the historical data for the arrival time of new tasks are generated with the combination of Gaussian distribution and uniform distribution for 3 hours juxtaposed together over 12 hours. The arrival times in the first and the last 3 hours window follow a Gaussian distribution, and the rest follow a uniform distribution. Each Gaussian distribution for arrival time follows the distribution with mean $\mu = 90$ minutes

Table 4.2: Distibution of number of tasks over 24 time periods

Time period	Distribution
0-6	$N_1 \sim U(50, 80)$
6-12	$N_2 \sim U(25 + 0.2N_1, 50 + 0.2N_1)$
12-18	$N_3 \sim U(30 + 0.2N_2, 50 + 0.2N_2)$
18-24	$N_4 \sim U(45 + 0.2N_3, 80 + 0.2N_3)$

and standard deviation $\sigma = 30$ minutes. The number of total tasks follows the distribution in table 4.2, where the U indicates the uniform distribution.

To train the neural network, a feed-forward neural network with ten hidden layers is used, and to obtain an output of the predicted new task arrivals in a particular time period for a particular grid, the information on new task arrivals for the preceding three time periods is used. The dataset is divided into $80-10-10$ for training, validation, and testing. Fig. 4.7 shows the error histogram of the trained and tested instances. It is clearly evident that the zero error in training has the highest peak in the maximum number of instances. The histogram also shows that very few instances have high errors, which may be unavoidable due to the random nature of the tasks.

Table 4.3: Reward accumulation during phase two

Number of tasks	**Max reward**	**Reward**	**Time needed (s)**
100	1480	1315.89	113
400	5160	4867.87	846
600	7240	6539.67	1055

The simulation is performed with a varied number of initial tasks known prior to the start of the mission, namely, 100, 400 and 600 tasks. Using the EMCGA-SS algorithm, these tasks are initially assigned to the 5 AEOSs before the mission

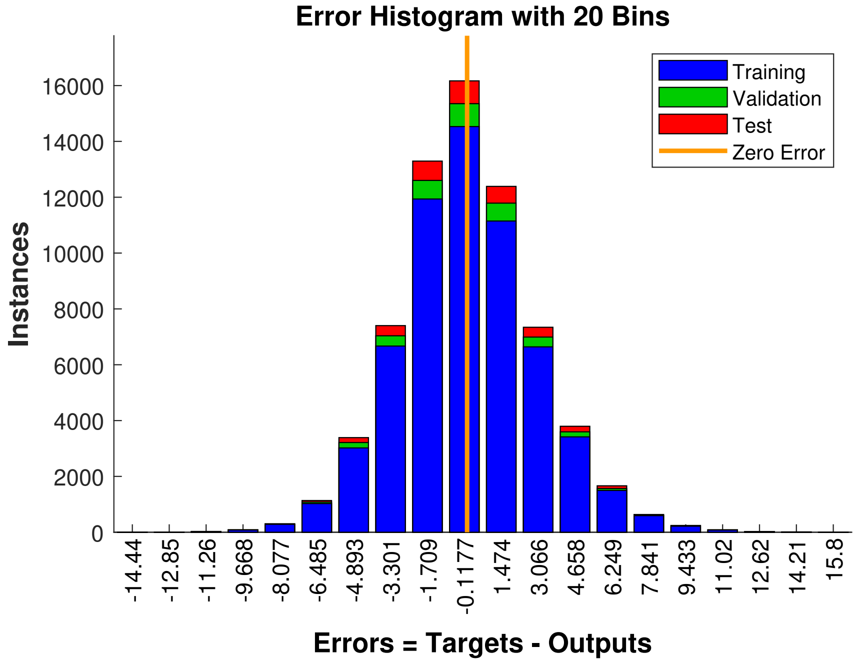


FIGURE 4.7: Error histogram for the performance of the neural network

starts. The potential rewards to be accumulated if all the tasks are successful are illustrated in Table 4.3. The table shows that the maximum reward is not achievable. This may occur due to the VTW of the tasks not overlapping with the trajectories of the satellites.

The table 4.4 illustrates the total reward achieved after the predictive schedule in phase three is complete. While generating the dummy tasks, the rewards for those are generated randomly between 1 to 5. The reward accumulated is lower for the original tasks from phase two due to the disruption in the schedule.

When new tasks arrive, during the mission, in the range of 100, 250 and 500 tasks during the mission, either a full reschedule at each time period during the

Table 4.4: Predicted reward during phase three

Number of original tasks	**Average number of predicted new tasks**	**Total reward**	**Reward for original tasks**	**Time needed (s)**
100	100	1562.54	1278.48	102
	250	1975.87	1254.23	134
	500	2659.65	1210.21	176
400	100	5231.16	4834.65	897
	250	5521.31	4798.21	932
	500	6128.97	4754.32	1044

Table 4.5: Comparison of the number of rescheduled tasks for the proposed method and complete reschedule

	Complete reschedule		**Proposed method**	
Number of new tasks	**Number of reschedule period**	**Number of tasks rescheduled**	**Number of reschedule period**	**Number of tasks rescheduled**
100	8	100	2	13
250	14	250	5	21
500	16	500	8	34

mission or the proposed method in an offline-online strategy may be applied. The new tasks might not arrive every time period, so rescheduling is only needed after those time periods when the new tasks arrive. Table 4.5 shows the number of rescheduled periods and the number of tasks to be rescheduled for full rescheduling and the proposed strategy.

Table 4.6 illustrates the rewards accumulated from using the proposed method over rescheduling. The table also shows the reward accumulated if the final rescheduling of phase four of the proposed method is skipped. The table further shows the reward accumulated when the complete rescheduling is done with minimum disruption as an objective. Finally, the reward achieved by simply placing the tasks in the idle times of the original schedule is also compared in the table.

Table 4.6: Comparison of total reward achieved with strategies including the proposed method and complete rescheduling

Number of original task	**Number of new task**	**Max reward**	**With rescheduling**	**Reschedule with minimum disruption**	**Proposed method**	**Proposed method without reschedule**	**Assigning to idle times**
100	100	2740	2235.46	2145.56	2167.26	1976.35	1689.81
	250	5060	4512.35	4311.52	4396.81	4153.87	3796.45
	500	10120	8986.22	8289.64	8661.49	8396.51	7978.17
400	100	8130	6980.23	6685.67	6741.32	6481.79	6134.53
	250	11150	9655.48	9257.48	9386.15	9143.58	8749.82
	500	15640	12896.76	12474.49	12564.54	12197.83	11946.73

One of the most important requirements of an online scheduler is the fast processing time during the mission. Table 4.7 shows the total processing time and the processing time during the mission for the above-mentioned strategies. The table clearly illustrates that the proposed method with skipping the rescheduling and placing the tasks into idle spaces is the quickest during the mission.

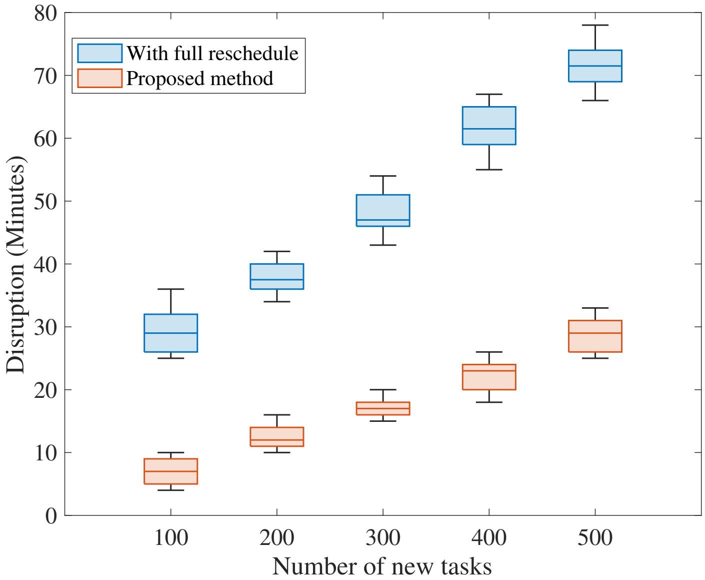


FIGURE 4.8: Disruption in minutes vs. number of new tasks over 100 Monte-Carlo runs

In Fig. 4.8, a boxplot is depicted to illustrate the increase in disruption to the existing schedule when new tasks arrive during the mission. This boxplot is generated with 100 Monte-Carlo runs. The figure shows that with the proposed method, the disruption to the existing schedule would be much lower than rescheduling at every instance when new tasks arrive. This clearly shows a completely online strategy is harmful and will create delays in task completion and will affect the reward collection, as a whole. A combination of an online-offline strategy is a much

Table 4.7: Comparison of processing time with strategies including the proposed method and complete rescheduling

		With rescheduling		**Proposed method**		**Proposed method without reschedule**		**Reschedule with minimum disruption**		**Assigning to idle times**	
Number of original tasks	**Number of new tasks**	**Total processing time**	**During mission processing time**	**Total processing time**	**During mission processing time**	**Total processing time**	**During mission processing time**	**Total Processing time**	**During mission processing time**	**Total processing time**	**During mission processing time**
100	100	742	631	436	297	218	58	813	698	167	54
	250	1750	1612	785	658	267	76	1986	1805	191	78
	500	3112	2984	1046	867	319	97	3357	3114	208	95

beneficial choice in dynamic scenarios such as new task arrivals during ongoing missions.

4.6 Conclusions

This work focuses on accommodating newly arrived tasks during the mission to the existing schedule of AEOSs. Although a complete reschedule on a regular interval is an easier option, it creates disruptions to the existing schedule and has a high computational cost. A four-phase, online-offline AEOS scheduling strategy is proposed in this work to deal with new task arrivals. The historical pattern of the new task arrivals is learned using a neural network-based function approximator in the first phase. In the second phase, pre-mission user-requested tasks are scheduled using a constrained optimization problem with resource constraints. The third phase uses the obtained schedule from the previous phase and the trained data from phase one to create a predictive schedule with dummy new task arrivals. While these three above-mentioned phases are performed before the mission, during the mission, the fourth phase assigns the newly arrived real tasks to the dummy tasks in the predictive schedule using a simple linear assignment problem. Only a few tasks, which cannot be assigned, are kept for rescheduling. The experimental results show that although a complete reschedule might yield more reward in most cases, the computational cost and the disruption to the existing schedule are too high to perform in an online manner with respect to the proposed method.

A possible future scope of this work is extending an online-offline strategy for accommodating new task arrivals when the AOIs are large and cannot be imaged in a single scan, or there are multiple visit requirements for the user-requested tasks.

Bibliography

Chatterjee, A. and Tharmarasa, R. (2022). Reward factor-based multiple agile satellites scheduling with energy and memory constraints, *IEEE Transactions on Aerospace and Electronic Systems* **58**(4): 3090–3103.

Chu, X., Chen, Y. and Tan, Y. (2017). An anytime branch and bound algorithm for agile earth observation satellite on-board scheduling, *Advances in Space Research* **60**(9): 2077–2090.

Deb, K. (2011). *Multi-objective optimisation using evolutionary algorithms: an introduction*, Springer.

Doerr, B., Le, H. P., Makhmara, R. and Nguyen, T. D. (2017). Fast genetic algorithms, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 777–784.

Elfwing, S., Uchibe, E. and Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Networks* **107**: 3–11.

Ferrari, S. and Stengel, R. F. (2005). Smooth function approximation using neural networks, *IEEE Transactions on Neural Networks* **16**(1): 24–38.

Globus, A., Crawford, J., Lohn, J. and Pryor, A. (2003). Scheduling earth observing satellites with evolutionary algorithms, *International conference on space mission challenges for information technology*.

Habet, D., Vasquez, M. and Vimont, Y. (2010). Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite, *Computational optimization and applications* **47**: 307–333.

Haijiao, W., Zhen, Y., Wugen, Z. and Dalin, L. (2019). Online scheduling of image satellites based on neural networks and deep reinforcement learning, *Chinese Journal of Aeronautics* **32**(4): 1011–1019.

Hall, N. G., Liu, Z. and Potts, C. N. (2007). Rescheduling for multiple new orders, *INFORMS Journal on Computing* **19**(4): 633–645.

Hall, N. G. and Magazine, M. J. (1994). Maximizing the value of a space mission, *European journal of operational research* **78**(2): 224–241.

Hall, N. G. and Potts, C. N. (2004). Rescheduling for new orders, *Operations Research* **52**(3): 440–453.

He, L., Liu, X.-L., Chen, Y.-W., Xing, L.-N. and Liu, K. (2019). Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment, *Advances in Space Research* **63**(2): 897–912.

Kuhn, H. W. (2010). The hungarian method for the assignment problem. in 50 years of integer programming 1958-2008, *Springer* **6**: 29–47.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M. and Bataille, N. (2002). Selecting and scheduling observations of agile satellites, *Aerospace Science and Technology* **6**(5): 367–381.

Liu, X., Laporte, G., Chen, Y. and He, R. (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time, *Computers & Operations Research* **86**: 41–53.

Lu, J., Chen, Y. and He, R. (2020). A learning-based approach for agile satellite on-board scheduling, *IEEE Access* **8**: 16941–16952.

Peng, G., Song, G., He, Y., Yu, J., Xiang, S., Xing, L. and Vansteenwegen, P. (2020). Solving the agile earth observation satellite scheduling problem with time-dependent transition times, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52**(3): 1614–1625.

Peng, S., Chen, H., Du, C., Li, J. and Jing, N. (2018). Onboard observation task planning for an autonomous earth observation satellite using long short-term memory, *IEEE Access* **6**: 65118–65129.

Scarselli, F. and Tsoi, A. C. (1998). Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results, *Neural networks* **11**(1): 15–37.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*, MIT press.

Svozil, D., Kvasnicka, V. and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks, *Chemometrics and intelligent laboratory systems* **39**(1): 43–62.

Tangpattanakul, P., Jozefowiez, N. and Lopez, P. (2015). A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite, *European Journal of Operational Research* **245**(2): 542–554.

Wang, X., Song, G., Leus, R. and Han, C. (2019). Robust earth observation satellite scheduling with uncertainty of cloud coverage, *IEEE Transactions on Aerospace and Electronic Systems* **56**(3): 2450–2461.

Wang, X., Wu, G., Xing, L. and Pedrycz, W. (2020). Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions, *IEEE Systems Journal* .

Wu, G., Wang, H., Pedrycz, W., Li, H. and Wang, L. (2017). Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy, *Computers & Industrial Engineering* **113**: 576–588.

Yang, S., Ting, T., Man, K. L. and Guan, S.-U. (2013). Investigation of neural networks for function approximation, *Procedia Computer Science* **17**: 586–594.

Zhai, X., Niu, X., Tang, H., Wu, L. and Shen, Y. (2015). Robust satellite scheduling approach for dynamic emergency tasks, *Mathematical Problems in Engineering* **2015**.

Chapter 5

Conclusions

5.1 Research Summary

In this thesis, AEOS scheduling problems with realistic operational constraints and task specifications are studied. A mixed-integer non-linear optimization problem is formulated in Chapter 2 that finds the optimal schedule by maximizing the reward with realistic satellite resources as energy and memory constraints. To incorporate the need for multiple scans to complete a task, a reward factor is included in the objective function. Probability-based failure and success rates for completion of scanning the tasks are also taken into consideration. An elitist mixed-coded genetic algorithm-based methodology has been developed to solve the proposed scheduling model. A three-stage AEOS scheduling model has been proposed to find the optimal task assignment when the AOIs are too large to complete in a single scan in Chapter 3. The proposed methodology minimizes the region overlapping between the scans, which is a drawback of strip-based task segregation for larger task regions. In both small and large-scale scenarios, this has been illustrated with simulation results. The inclusion of MDP in dealing with task failure uncertainty provides superior performance in terms of reward accumulation. In Chapter 4, new task arrivals during the mission are accommodated in the existing schedule of AEOSs which is generated prior to the mission. Although a complete reschedule on a regular interval is an easier option, it creates disruptions to the existing schedule and has a high computational cost. A four-phase, online-offline AEOS scheduling strategy is proposed in this chapter.

5.2 Future Scopes of Research

An obvious future step of this work is developing an online-offline strategy for accommodating new task arrivals when the AOIs are large and cannot be imaged in a single scan, or there are multiple visit requirements for the user-requested tasks. Consideration of the AEOSs as a decentralized network, where the satellites are autonomous agents that are cooperating with each other can be an interesting future step as well.