Integrated Scheduling of EOS And Ground Stations Using Genetic Algorithm and Population Based Iterated Local Search

A project report submitted

in partial fulfilment for award of the Degree of

Bachelor of Technology

in

Aerospace Engineering

by

Ajaya Kumar Patel
SC19B002

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

THIRUVANANTHAPURAM

January - May 2023

CERTIFICATE

This is to certify that the thesis titled '**Integrated Scheduling of Earth Observation Satellites and Ground Stations Using Genetic Algorithm and Population Based Iterated Local Search**', submitted by **Ajaya Kumar Patel**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, for the award of the degree of **B.Tech** in **Aerospace Engineering**, is a bonafide record of the research work done by **him** under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Girish B.S.
Associate Professor
Department of Aerospace Engineering

Place: IIST, Thiruvanathapuram
May 2023

Dr. Aravind Vaidyanathan
Head of the Department
Department of Aerospace Engineering

Declaration

I declare that this thesis titled '**Integrated Scheduling of Earth Observation Satellites and Ground Stations Using Genetic Algorithm and Population Based Iterated Local Search**' submitted in fulfillment of the Degree of **Bachelor of Technology in Aerospace Engineering** is a record of original work carried out by me under the supervision of **Dr. Girish B.S.**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited

Place: IIST, Thiruvanathapuram
May 2023

Ajaya Kumar Patel
SC19B002

Acknowledgements

I would like to express my heartfelt gratitude to my professor Dr. Girish B.S. and our HOD Dr. Aravind Vaidyanathan for providing me with the beautiful opportunity to work on this project. Not only did this assist me in finishing this project, but it also introduced me to many new things.

Ajaya Kumar Patel

Abstract

The scheduling of Earth observation satellites is a complex task that involves balancing multiple constraints to optimize resource utilization. In this project, a novel approach is proposed for optimizing the scheduling of up to eight real-time satellites and up to 150 randomly located targets in India, with a particular focus on spot targets. The proposed approach involves using two cases, one with Genetic Algorithm (GA) and the other with Population-based Iterative Local Search (PBIL).

Objective evaluation is done using a resources constraint longest path algorithm (based on Dijkstra's algorithm), taking into account various constraints such as visibility window, transition time, memory, energy, and the spot target. Additionally, we include ground station download scheduling, which has not been considered in previous studies. Priority and quality of scheduling are optimized, with priority assigned based on the importance of the target and three priority levels (100, 10, and 1). The quality of the target is between 0.7 and 1, depending on the roll and pitch angles at which the target is captured.

The results demonstrate that PBIL provides superior solution quality to GA, despite taking longer to converge. Furthermore, the inclusion of ground station download scheduling enhances the overall efficiency of the scheduling process.

The proposed approach provides an efficient and effective solution to Earth observation satellite scheduling while considering various constraints and optimizing both priority and quality, with a focus on spot targets. The formulation of the scheduling problem discussed in this research is based on the specific requirements and details provided by the Mission Planning and Design Group at the U R Rao Satellite Centre (URSC), ISRO. Future research could focus on optimizing the objective function further and enhancing the efficiency of both the Genetic Algorithm (GA) and Population-based Iterative Local Search (PBIL) algorithms, for continued improvements in Earth observation satellite scheduling.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

In recent years, the demand for earth observation data has increased significantly as decision-makers in both the public and private sectors recognize its potential to make critical decision-making processes. This data can be used to monitor changes in the environment, track natural disasters, make urban planning, support agriculture, and many other applications.

However, to obtain this data, a complex system of earth observation satellites and ground stations must work together seamlessly. Scheduling these resources effectively can be challenging, and inefficiencies in the scheduling process can lead to missed opportunities to gather critical data.

Therefore, the main motivation for this research is to develop a more efficient and integrated system for scheduling Earth observation satellites and ground stations. By doing so, we can increase the amount and quality of data gathered, leading to more informed decision-making across sectors.

The benefits of integrated scheduling of earth observation satellites and ground stations are many. For the government, this research can lead to improved disaster response and management, better monitoring of critical infrastructure, and more effective natural resource management. For the private sector, it can lead to improved forecasting, better supply chain management, and more efficient use of resources. For example, satellite imagery can be used to track the movement of goods and shipments across borders, monitor inventory levels, and even assess the condition of crops and natural resources that are used in manufacturing processes.

In addition, the research can benefit society as a whole by providing critical information for climate change research, improving public health and safety, and supporting efforts to protect the environment.

1.2 Earth observation satellites

Earth observation satellites are spacecraft designed to orbit the Earth and gather data and imagery of the planet's surface and atmosphere. These satellites use a range of technologies to observe the Earth, including remote sensing instruments, cameras, and radar systems.

Earth observation satellites provide a wealth of data that can be used for a wide range of applications. One of the primary uses of these satellites is for weather forecasting and climate research. Satellites can gather data on temperature, humidity, air pressure, and other atmospheric variables, which can be used to predict weather patterns and monitor climate change.

In addition to weather and climate applications, earth observation satellites can be used for a range of other purposes, including:

Environmental monitoring: Satellites can monitor changes in land use, deforestation, and other environmental factors that impact the health of the planet.

Agriculture: Satellites can provide data on soil moisture levels, crop health, and other agricultural factors, which can be used to improve farming practices and increase crop yields.

Disaster management: Satellites can be used to monitor natural disasters, such as hurricanes, floods, and wildfires, and provide data that can help emergency responders and aid organizations.

Urban planning: Satellites can provide data on population density, land use, and other factors that can be used to inform urban planning decisions.

Earth observation satellites are typically placed in either low Earth orbit (LEO) or geostationary orbit (GEO). LEO satellites orbit the Earth at an altitude between 200 and 2000 kilometers and can cover the entire planet in just a few days. GEO satellites, on the other hand, orbit the Earth at an altitude of 36,000 kilometers and remain stationary over a fixed point on the Earth's surface. This makes them ideal for applications such as weather forecasting and communication. Indian active satellite and there properties are shown in the Table 1.1.

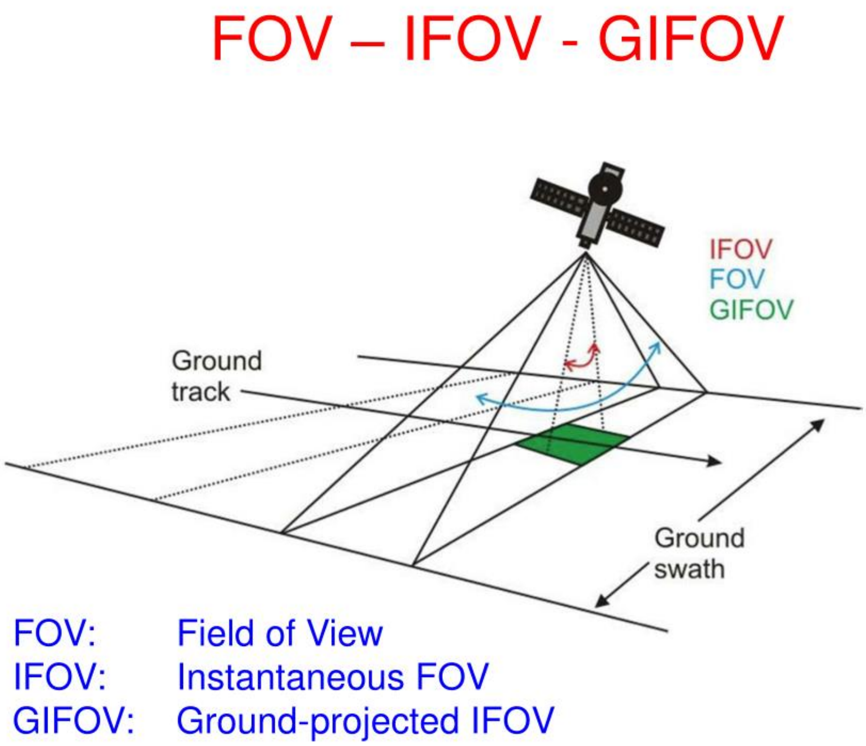

Fig. 1.1 Terms associated with satellite-imaging [1]

The terms (Fig 1.1) used in satellite imaging are as follows:

- **Nadir Point:** The point directly below a satellite.
- **Ground track:** The ground track is the path that a satellite follows over the surface of the Earth as it orbits. It is the projection of the satellite's orbit onto the Earth's surface, and it can be used to predict when and where the satellite will pass over a particular location on the Earth.
- **IFOV:** Instantaneous field of view is area that its sensor can image at instant.
- **FOR:** The total area that a satellite can observe. It is an important factor in mission planning and scheduling, as it determines the satellite's capability to capture comprehensive data and provide detailed observations.
- **Roll limits** $[\phi^-, \phi^+]$: The maximum angle of rotation around its longitudinal axis
- **Pitch Limits** $[\theta^-, \theta^+]$**:** The maximum angle of rotation around its transverse axis

Table 1.1 Operational Indian Earth Observation Satellites. Source: [4]

Satellite	Inclination (deg)	Apogee (km)	Perigee (km)	Period (min)	Orbit	Payload	Type	
							signal	Sensor
CARTOSAT	97.83	623	621	96.99	SSO	PAN	Visible light	Passive
CARTOSAT 2	97.94	639	636	97.32	SSO	PAN	Visible light	Possive
CARTOSAT 2A	97.95	648	628	97.32	SSO	PAN	Visible light	Passive
CARTOSAT 2B	97.94	647	628	97.32	SSO	PAN	Visible light	Passive
CARTOSAT 2C	97.47	522	503	94.72	SSO	PAN	Visible light	Passive
						HRMX	Visible light	Passive
						EMM	Visible light	Passive
CARIOSAT 2D	97.37	518	506	94.72	SSO	PAN	Visible light	Passive
						HRMX	Visible light	Passive
						EvM	Visible light	Passive
CARTOSAT 2E	97.44	518	506	94.72	SSO	PAN	Visible light	Passive
						HRMX	Visible light	Passive
						EvM	Visible light	Passive
CARTOSAT 2F	97.41	515	510	94.72	SSO	PAN	Visible light	Passive
						HRMX	Visible light	Passive
						EvM	Visible light	Passive
CARTOSAT 3	97.52	525	502	94.75	SSO	PAN	Visible light	Passive
						MX	Visible light	Passive
HYSIS	97.92	645	630	97.32	SSO	HySIS	Visible light	Passive
MEGHA-TROPIQUES	19.98	880	866	102.26	LEO	MADRAS	Microwave	Passive
						ROSA	Microwave	Passive
						SAPHIR	Microwave	Passive
						ScaRaB	Visible light	Passive
OCEANSAT 2	98.3	729	725	99.19	SSO	OSCAT	Microwave	Passive
						OCM	Visible light	Passive
						ROSA	Microwave	Passive
RESOURCESAT	98.5	825	740	100.35	SSO	AWiFS	Visible light	Passive
						LISS-4	Visible light	Passive
						LISS-3	Visible light	Passive
RESOURCESAT 2	98.7	828	820	101.23	SSO	AWiFS	Visible light	Passive
						LISS-4	Visible light	Passive
						LISS-3	Visible light	Passive
RESOURCESAT 2A	98.61	825	825	101.23	SSO	AWiFS	Visible light	Passive
						LISS-4	Visible light	Passive
						LISS-3	Visible light	Passive
RISAT	97.57	544	537	95.3	SSO	SAR-C	Microwave	Active
RISAT 2	41.21	461	454	93.6	LEO	SAR-X	Microwave	Active
RISAT 2B	37	584	577	96.13	LEO	SAR-X	Microwave	Active
RISAT 2BRl	36.97	589	579	96.21	LEO	SAR-X	Microwave	Active
SCATSAT	98.18	731	722	99.19	SSO	oscxr	Microwave	Active

In satellite imaging operation two type of satellite is used. One is agile and other one is non-agile. Agile satellites are those that can rapidly change their orientation or pointing direction in space. These satellites typically have a high degree of maneuverability and can quickly reposition their sensors to capture data over a wide range of targets or areas of interest. Non-agile satellites, on the other hand, have a fixed orientation or pointing direction in space. These satellites typically have a limited ability to maneuver and are designed to observe a specific area or target over an extended period of time. In recent time agile is used mainly. Agile and Non-Agile satellite are shown in the Fig 1.2.

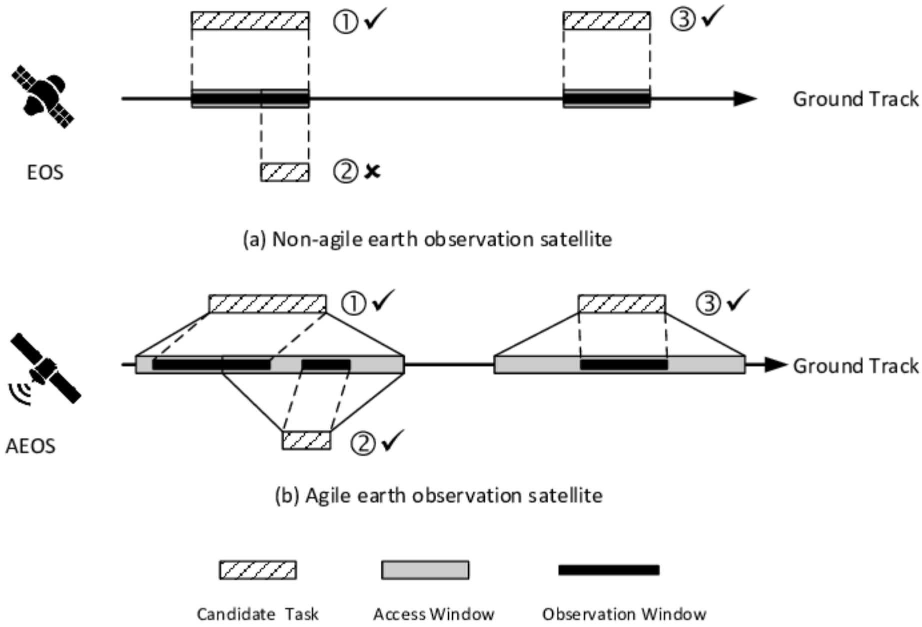

Fig. 1.2 Non-Agile and Agile Satellites [1]

1.3 Download Process

Satellites utilize antennas to capture and transmit images. These antennas can either be fixed directly onto the satellite's body or connected through a gimbal system, which allows for directional adjustment. Antennas can have two types of characteristics: uni-directional or omni-directional.

Omni-directional antennas have the ability to transmit data in all directions. However, due to their design, they can only transmit a limited amount of data. As a result, omni-directional antennas are commonly used for tasks such as telemetry and sending commands.

On the other hand, a focused unidirectional antenna is specifically employed to transmit large volumes of imaging data to ground stations. These antennas provide a concentrated and targeted transmission of data, ensuring efficient and high-speed transfer of the imagery.

To receive the imaging data transmitted by the satellite, ground stations are equipped with large antennas that are capable of tracking the satellite's movement as it passes over the station. These tracking antennas are designed to follow the satellite's trajectory accurately, allowing for a continuous and uninterrupted data reception.

Similar to the process of capturing images, the transmission of data from satellites to ground stations occurs during specific time intervals known as download windows. These download windows are predetermined periods during which the satellite and the ground station are in optimal alignment for efficient data transfer.

In summary, the use of antennas plays a crucial role in the transmission of images captured by satellites. Omni-directional antennas are suitable for low-data tasks such as telemetry and commands, while focused unidirectional antennas enable the high-volume transmission of imaging data. Ground stations employ large tracking antennas to receive the transmitted data, and the timing of these data transfers is coordinated through predetermined download windows.

1.4 Integrated Scheduling Problem

The integrated scheduling problem involves the coordination of both imaging and downloading activities for targets within a given time horizon. We are provided with information about the satellites, targets, and their respective priorities. The objective is to create a schedule for each satellite during the specified time period that maximizes the total priority of the targets that are both imaged and downloaded.

Creating a satellite schedule entails determining precisely what each satellite should do and when. This involves deciding when a satellite will capture images of specific targets and when it will download data to the corresponding ground station.

One of the key challenges in this problem is the efficient utilization of limited resources, such as satellite energy and memory capacity. It is crucial to optimize the use of these resources while ensuring that data is acquired at the right time and with the required level of accuracy. This necessitates striking a balance between competing demands and priorities. To tackle this challenge, advanced algorithms are developed to facilitate effective resource allocation.

Chapter 2

Literature review

The scheduling of Earth Observation Satellites (EOS) plays a critical role in maximizing the efficiency and effectiveness of data collection and observation tasks. The integration of various scheduling elements, such as target selection, satellite allocation, and ground station coordination, poses unique challenges that have attracted significant attention in the literature.

This literature review explores the existing research and developments in the field of integrated scheduling for EOS. The objective is to gain insights into the problem types, scheduling approaches, and constraints considered by researchers. By examining these aspects, valuable gaps, potential opportunities, and promising directions for future research can be identified.

The review begins by categorizing the targets under observation into two main types: spot targets and polygon targets. This categorization provides valuable insights into the considerations and approaches taken in scheduling tasks. Subsequently, the scheduling approaches based on the configuration of satellites involved are explored, distinguishing between single-satellite scheduling and multi-satellite scheduling.

Moreover, the review examines the constraints that have been considered in satellite scheduling. These constraints encompass factors such as the visibility window, transition time, memory, and energy. Understanding these constraints is crucial for developing scheduling algorithms that optimize the allocation of resources while adhering to operational limitations.

The problem formulation and solution strategies will be explained in detail in the following chapters. By reviewing the existing literature, this study aims to contribute to the understanding and advancement of integrated scheduling for EOS, with implications for improved resource allocation, enhanced data collection, and efficient utilization of satellite systems.

2.1 Target Categorization

In the literature on integrated scheduling, researchers have explored various problem types based on the type of target being observed. To categorize the targets, two main types have been identified: spot targets and polygon targets. The distinction between these two types provides insights into the considerations and approaches taken in scheduling tasks.

2.1.1 Spot Targets:

Spot targets refer to small areas of interest that require observation. They typically encompass circular areas with a radius of fewer than 10 km [6]. Researchers have addressed the scheduling of spot targets in multiple studies [6, 7, 8, 9, 10, 11], focusing on optimizing the allocation of resources and maximizing the quality and coverage of observations.

2.1.2 Polygon Targets:

Polygon targets involve larger areas, such as towns or cities, and are characterized by their polygonal shape. The length of these polygons typically ranges from 20 to 100 km [6]. Several research works have specifically tackled the scheduling challenges associated with polygon targets [9, 12, 13, 14]. These studies aim to develop effective strategies to allocate resources and optimize the scheduling of observations for these larger geographical areas.

By categorizing targets into spot and polygon types, researchers have gained a deeper understanding of the scheduling requirements and constraints associated with each target category. This categorization helps guide the development of tailored scheduling algorithms and techniques for integrated scheduling in the context of Earth Observation Satellites (EOS) and ground stations.

2.2 Scheduling Approaches Based on Configuration

The literature on integrated scheduling has also examined the number of satellites involved in the scheduling process, leading to the identification of two main scheduling types: single-satellite scheduling and multi-satellite scheduling. These scheduling types provide insights into the complexities and considerations associated with coordinating observations from different satellite configurations.

2.2.1 Single-Satellite Scheduling:

Single-satellite scheduling involves the coordination of observations from a single satellite. This scheduling type has been the focus of several studies [6, 7, 9, 15, 16, 17]. Researchers in this area aim to optimize the scheduling of observations from a single satellite, considering various constraints and objectives.

2.2.2 Multi-Satellite Scheduling:

Multi-satellite scheduling, on the other hand, deals with the coordination of observations from multiple satellites. In recent literature, there has been an increasing emphasis on the challenges and opportunities presented by this type of scheduling [13, 18, 19, 20, 21]. Researchers recognize that leveraging the capabilities of multiple satellites can offer improved coverage, increased revisit rates, and enhanced overall scheduling efficiency.

Recent papers have particularly focused on addressing the multi-satellite scheduling problem [10, 22, 23, 24]. For example, Cho et al. [24] have investigated the optimization of satellite constellations. A satellite constellation refers to a group of satellites working together to provide specific services or applications. These constellations are often positioned in a specific pattern and orbit around the Earth at a similar altitude. The optimization of satellite constellations presents unique challenges in terms of coordination, resource allocation, and mission objectives.

One notable approach in multi-satellite observation scheduling is presented by Xu et al. [13]. They specifically address the scheduling of observations for very large area targets involving multiple satellites. Their methodology incorporates a three-phase framework to achieve optimal scheduling outcomes. The first phase involves discretizing the target area to establish an evaluation system. In the second phase, the targets are decomposed into strips, and corresponding visibility windows are calculated. Finally, a genetic algorithm is employed during the scheduling phase to optimize the observation schedule based on an objective function proportional to the fraction of the area observed.

Janakiram [25] focuses on solving the agile satellite scheduling problem and explores three distinct methods: two variations of population-based Iterated Local Search (P-ILS) and a hybrid Particle Swarm Optimization-Local Search (PSO-LS). Through a comparative analysis, it was observed that P-ILS-2 and PSO-LS yield similar results in terms of both objective value and computation time, demonstrating their effectiveness. However, P-ILS-1 exhibited comparatively lower performance among the three approaches.

2.3 Constraints Considered in Scheduling

Satellite scheduling algorithms play a crucial role in optimizing observations while taking into account various operational constraints. The literature on satellite scheduling has identified and addressed several constraints to ensure effective scheduling outcomes. This subsection discusses key constraints that have been considered in satellite scheduling research.

One important constraint is the visibility window [22]. The visibility window refers to the duration during which a target is visible to the satellite. Maximizing the time that a satellite can spend observing a target is essential, and the visibility window constraint helps determine the optimal scheduling of observations.

Another significant constraint is the transition time [23]. Transition time refers to the time required for a satellite to move from one target to another. Minimizing the transition time is crucial to optimize the utilization of satellite resources by reducing the time spent in transitioning between targets.

Memory and energy constraints are two additional factors that have received considerable attention in the literature [13, 14, 15, 21, 23, 24, 26, 27]. Memory constraints pertain to the amount of onboard storage capacity available on the satellite for data storage, while energy constraints focus on managing the energy resources required for satellite operations. Effectively considering these constraints is vital for ensuring efficient utilization of satellite resources.

Cloud cover is another significant constraint that has been taken into account in satellite scheduling research [14]. Cloud cover directly impacts image quality, making it necessary to consider this constraint to optimize observation outcomes.

Stereo imaging, which involves capturing images of the same target from multiple angles, is another constraint that has been considered in the literature. This technique enhances the quality and accuracy of captured images, leading to improved data analysis and interpretation.

In addition, image quality and field-of-regard (FOR) limits have been identified as important constraints in satellite scheduling. Image quality constraint, as studied by Liu et al. [28], emphasizes the need to maintain high-quality images for effective analysis. Field-of-regard limit constraint involves defining the angular range within which a satellite can observe targets.

Considering these constraints in satellite scheduling algorithms ensures that observations are optimized while adhering to various operational limitations.

2.4 Solution Strategies in Scheduling

The literature on satellite scheduling has explored various solution strategies to tackle the complexity of scheduling problems. This subsection discusses different solution strategies that have been employed to address satellite scheduling challenges effectively.

One commonly used strategy is Ant Colony Optimization (ACO), which takes inspiration from the foraging behavior of ants [29]. ACO algorithms have been applied to satellite scheduling problems, leveraging the decentralized decision-making and pheromone communication mechanism of ants to find optimal schedules.

Another approach is Differential Evolution (DE), which is a population-based optimization algorithm [29]. DE generates new solutions by combining information from existing solutions, allowing for exploration and exploitation of the search space. It has shown promising results in solving satellite scheduling problems.

Genetic Algorithms (GA) have also been successful in solving complex satellite scheduling problems that involve multiple satellites and various constraints [13, 16, 30, 31]. GAs are known for their ability to search large solution spaces and find high-quality solutions. They employ evolutionary operators such as selection, crossover, and mutation to iteratively improve the schedules.

Simulated Annealing (SA) is a stochastic optimization algorithm that has been utilized in satellite scheduling [26]. It uses a probabilistic criterion to accept or reject new solutions during the search process, allowing it to escape local optima and explore the solution space more effectively.

In addition to these metaheuristic algorithms, other solution techniques such as mathematical programming methods (e.g., branch and bound, constraint programming, and linear programming), enumeration methods, greedy methods, local search, and mixed-integer linear programming have also been applied in satellite scheduling research [7, 15, 18, 23, 32]. These techniques leverage mathematical optimization principles and search algorithms to find optimal or near-optimal solutions to the scheduling problems.

2.5 Integrated Scheduling

The field of integrated scheduling for Earth Observation Satellites (EOS) and ground stations, has seen relatively fewer studies compared to the extensive research focused solely on observation scheduling. Most of the existing literature primarily emphasizes scheduling imaging tasks without incorporating download scheduling. But in recent years, there has been a discernible growth in research that tackles the integrated scheduling problem, employing diverse methodologies and addressing specific challenges.

One early study by Wang et al. [33] examined the integrated scheduling problem. They formulated a nonlinear model that incorporated complex satellite operational constraints and request preferences such as visibility time windows, transition time between consecutive observations or downloads, memory and energy capacity, polygon target requests, and priorities. To solve the problem efficiently, they developed a priority-based heuristic algorithm with conflict-avoidance.

Augenstein et al. [32] proposed a novel approach using mixed-integer linear programming (MILP) for scheduling a large constellation of Earth-imaging satellites. Their methodology involved dividing the integrated scheduling problem into two distinct parts: download window scheduling and observation scheduling. Initially, they focused on download window scheduling by assigning priority values to different download windows based on the number of targets the satellite passed over just before each window started. This prioritization strategy aimed to optimize the utilization of available download opportunities. Once the download windows were scheduled, the authors proceeded to schedule the target imaging tasks in a way that maximized the total priority of all imaged targets. For download window scheduling, they formulated a MILP model, while for observation scheduling, they employed a dynamic programming method. By dividing the problem into these two stages, Augenstein et al. aimed to efficiently allocate download resources and optimize the overall scheduling of imaging tasks.

Qi et al. [34] introduced a comprehensive multi-EOS scheduling scheme that encompassed two key stages: mission pre-planning and mission re-planning. Their approach considered crucial constraints such as energy and ground station availability. In the initial stage, they devised an Evolutionary Ant Colony Optimization (EACO) method to generate optimal solutions for the mission schedule. By leveraging the advantages of both evolutionary algorithms and ant colony optimization, this stage aimed to produce high-quality schedules. Subsequently, the second stage employed an interactive re-planning method to further refine the mission schedule. This iterative process allowed for adjustments and improvements to be made based on the evolving operational environment.

Zhang and Xing [35] formulated the integrated scheduling problem as an MILP and solved it using an improved genetic algorithm. They considered memory, energy, and ground station constraints in their model. Further Wen et al. [36] addressed the single-satellite integrated scheduling problem using a time continuous model. They employed a hybrid Actor-Critic reinforcement learning method to solve the problem efficiently.

The literature review highlights the existence of different approaches to addressing the integrated scheduling problem, with a focus on operational constraints such as transition time, energy, and memory. However, an aspect that has often been overlooked is the proper consideration of ground station operational constraints. Furthermore, some studies rely on download schemes that are based on assumptions and may not always yield optimal results.

For example, Zhang and Xing [35] utilize a genetic algorithm to allocate a satellite and a ground station to each target, followed by the Minimum Start Time (MST) strategy for determining imaging time and the Earliest Feasible Time (EFT) strategy for finding the download time. However, these strategies may not take into account all relevant factors and may not lead to the most efficient scheduling outcomes.

In this work, the focus is on spot targets, and no assumptions are made regarding imaging and download strategies. Instead, a heuristic approach is employed to select imaging slots and download windows, followed by an exact method to determine the optimal schedule for the allocation. The download scheme prioritizes images based on their importance, ensuring that higher priority images are downloaded before lower priority ones. Additionally, various constraints including transition, energy, memory, and ground station limitations are taken into account.

Furthermore, it is important to mention that the formulation of the scheduling problem discussed in this research is based on the specific requirements and details provided by the Mission Planning and Design Group at the U R Rao Satellite Centre (URSC) - ISRO.

Chapter 3

Problem formulation

3.1 Problem Statement

The main goal of this project is to develop an optimal scheduling algorithm for a set of targets, satellites, and ground stations. The objective is to efficiently allocate the available resources (satellites and ground stations) to the targets in order to maximize the total weighted score of the targets imaged, weighted score is a function of priority and quality.

Let $T = T_1, T_2, \ldots, T_N$ be the set of N targets to be imaged. Each target T_i is characterized by its priority P_i, imaging duration d_i, and memory consumption M_i.

Let $S = S_1, S_2, \ldots, S_K$ be the set of K available satellites that can be used to image the targets. Each satellite S_k is characterized by its imaging capabilities such as spatial resolution, spectral resolution, and field of view.

Let $GS = GS_1, GS_2, \ldots, GS_L$ be the set of L available ground stations that can be used to receive the image data from the satellites. Each ground station GS_L is characterized by its receiving capabilities such as data rate and coverage area.

3.2 Satellite

In this project, the scheduling algorithm takes into account up to ten Agile satellites. Agile satellites are specifically chosen for their exceptional capabilities, including faster slew rates, which result in shorter transition times between observations. These satellites are actively operated by the Indian Space Research Organization (ISRO).

The selection of satellites is based on their imaging capabilities, which encompass crucial factors such as field of view. Satellites and their properties are shown in the table 1.1.

Table 3.1 Ground Station Location

ID	Location
GS_1	Bangalore
GS_1	Jodhpur
GS_1	Antarctica

3.3 Target

Up to 150 targets are taken into consideration. Each target is assigned a priority based on its level of importance. We have categorized the priorities into three levels: high, medium, and low, denoted by 100, 10, and 1, respectively.

The assignment of priorities allows us to effectively allocate the available resources to the targets. High-priority targets, indicated by a priority of 100, are of utmost significance and require immediate attention in terms of imaging. Medium-priority targets, with a priority of 10, hold a moderate level of importance. Lastly, low-priority targets, assigned a priority of 1, are comparatively less critical and can be scheduled accordingly.

By using multiple levels of priority,the scheduling algorithm aims to optimize the utilization of resources. It achieves this by prioritizing the imaging of targets based on their respective levels of importance. This approach facilitates the maximization of the total priority of imaged targets, ensuring an efficient utilization of available resources and the fulfillment of critical imaging requirements.

In addition, this project consider spot targets into the scheduling algorithm. Spot target are small area like small town, agricultural fields.

3.4 Ground Station

The download of data from the satellites is facilitated through the utilization of three ground stations. Each ground station is carefully assigned to a specific satellite during the scheduling process to ensure efficient data transfer. Location of graound station are shown in the table 3.1.

3.5 Objective:

The primary objective of the scheduling algorithm is to maximize the total weighted score of the targets imaged. The weighted score is a function of priority and quality, as shown in Equation 3.1. The priority of each target is defined by the parameter P_i. And the quality of each target is defined by the parameter Q_i. The quality of the images is depends on roll and pitch of the satellite.

$$\text{Weighted Score} = \text{Priority} \times \text{Quality} \tag{3.1}$$

3.5.1 Image Quality

An image quality parameter q_{ijkl} is defined for target T_i when it is imaged by satellite S_j in orbit O_{jk} at time slot t_l. It depends on the relative position of the target with respect to the satellite when it is being imaged.

To formulate the image quality parameter, various parameters such as pitch and roll are taken into account. The parameters defined in equations 3.2 and equation 3.3 are formulated such that a target at the nadir point will get the highest value 1 and a target at the extremes of FOR will get the least value 0.7.

The image quality parameter Q_{ijkl} is a weighted sum of q^{θ} and q^{ϕ}, which are the parameters for pitch and roll, respectively. The weights for pitch and roll, denoted by κ_{θ} and κ_{ϕ} mentioned in eq 3.4. The image quality equation is similar to the ones proposed by He et al [11] and Liu et al [28].

$$q^{\theta}_{ijkl} = \begin{cases} \left(10 - 9\frac{\theta_{ijkl}}{\widehat{\theta}^{+}_{j}}\right) & \text{if } \theta \geq 0 \\ \left(10 - 9\frac{\theta_{ijkl}}{\widehat{\theta}^{-}}\right) & \text{if } \theta < 0 \end{cases} \tag{3.2}$$

$$q^{\phi}_{ijkl} = \begin{cases} \left(10 - 9\frac{\phi_{ijkl}}{\widehat{\phi}^{+}}\right) & \text{if } \phi \geq 0 \\ \left(10 - 9\frac{\phi_{ijkl}}{\widehat{\phi}^{-}}\right) & \text{if } \phi < 0 \end{cases} \tag{3.3}$$

$$Q_{ijkl} = \kappa_{\theta} q^{\theta}_{ijkl} + \kappa_{\phi} q^{\phi}_{ijkl} \quad \text{where } \kappa_{\theta} + \kappa_{\phi} = 1 \tag{3.4}$$

Here, i is the target index, j is the satellite index, k is the orbit index and l is the time index.

3.6 Constraints

There are several constraints that need to be taken into consideration in the scheduling algorithm. These constraints are as follows:

3.6.1 Memory Constraint

Each target T_i is characterized by its memory consumption M_i. The total memory consumption of all targets imaged during a particular imaging session must not exceed the available memory capacity of the satellite. Therefore, the algorithm should allocate the targets to the available satellites in a way that ensures that the memory constraint is satisfied.

3.6.2 Energy Constraint

Each satellite S_k has a limited amount of energy that can be used to perform the imaging task. Therefore, the algorithm should ensure that the imaging plan is designed in a way that optimizes the use of the available energy resources to minimize the energy consumption.

3.6.3 Transition Time Constraint

After imaging a target, satellite undergoes a coordinated movement involving combined pitch and roll adjustments to precisely position itself for imaging the next target. The time needs for this is called Transition time. The transition time s_{ij} depends on the angle $_{ij}$ by which the satellite move from one target (i) to next target (j) and is also dependent on its angular speed. Equation 3.5 represent transition time between target i and target j, considering angular speed is constant.

$$s_{ij} = \frac{\phi_{ij}}{\omega} \tag{3.5}$$

3.6.4 Ground Station Coverage Constraint

The algorithm should ensure that the images received from the satellite are successfully transmitted to the ground station without any loss of data. Therefore, the algorithm should allocate the ground stations to the available satellites in a way that ensures that the data rate and coverage area constraints are satisfied.

Several other constraints are also considered in the scheduling algorithm. One such constraint is that a target can be imaged only once in the time horizon. This constraint ensures that the imaging resources are utilized efficiently and avoids repeated imaging of the same target.

3.7 Assumptions

1. Imaging and data downloading are mutually exclusive operations for a satellite, meaning it can either image or download data at any given time, but not both simultaneously.

2. Satellites are equipped with gimbal antennas that can rotate independently, eliminating the need for the satellites to orient themselves towards the Ground Station (GS).

3. The time interval between two consecutive imaging opportunities (i.e. transition time) is determined by the shortest angle between the satellite's orientations. The satellite maintains a constant angular velocity during rotation.

4. Download windows are either selected or not selected; partial selection of download windows is not considered.

5. Only point targets, rather than polygon targets, are taken into account for scheduling and imaging purposes.

6. Each Ground Station (GS) can receive data from only one satellite at any given time.

7. Between two satellite contacts, the GS antennas need to rotate to focus on the incoming satellites. To account for this delay, a minimum time gap of δT_G is always maintained between the end of one download and the start of the next download window for a GS.

8. Satellites are only capable of imaging during daytime.

9. Earth observation satellites rely on solar panels for their energy requirements. There is a dynamic balance between energy generation and consumption throughout the scheduling horizon, ensuring that the satellite generates sufficient energy to meet its power needs without exceeding its capacity. Although the amount of energy generated may vary based on the satellite's position relative to the Earth and the Sun, the energy generated in one orbit generally remains constant.

10. The targets of observation are assumed to be stationary and do not move during the observation period for simplicity purposes. Accounting for target movement would significantly increase the complexity of the scheduling problem. However, in reality, some targets may be moving, requiring additional considerations when scheduling the satellites.

11. The priority weights assigned to each target are assumed to be accurate and reflect their relative importance for observation. These weights may be subjective or subject

to change over time, but for the purpose of this scheduling problem, it is assumed that the assigned priority weights are reliable and accurately represent the importance of each target.

Chapter 4

Solution methodology

To solve the complex problem of scheduling earth observation satellites and ground stations, a solution methodology that utilizes two metaheuristics has been proposed. This methodology aims to optimize the use of limited resources and ensure that data is acquired at the right time and with the required level of accuracy. In this section, we will discuss the two metaheuristics that were used, namely the Genetic Algorithm (GA) and Population Based Iterated Local Search (PIBL), and how they were applied to the scheduling problem.

In this proposed solution methodology, two metaheuristics were used: Genetic Algorithm (GA) and Population Based Iterated Local Search (PIBL). GA is a population-based metaheuristic that emulates the natural selection process to evolve a population of candidate solutions towards better solutions. It randomly generates a set of potential solutions and evolves them over several generations through selection, crossover, and mutation operators. These operators generate new solutions that are evaluated using a fitness function and are selected based on their fitness value to become part of the new population. The process continues until a satisfactory solution is found or a termination criterion is met.

On the other hand, ILS is a trajectory-based metaheuristic that begins with an initial solution and iteratively improves it by applying local search in its neighborhood. PBIL searches the local optimum by iteratively perturbing the current solution and performing local search until a better solution is found or a termination criterion is met. It repeats this process until it converges to a satisfactory solution.

In this study, GA and PBIL are used separately to solve the proposed problem. The encoding scheme, initial population generation procedure, objective function, local search procedure, and termination criteria for each of these algorithms are explained in detail in the following sections.

Table 4.1 Targets allocated example problem

S_j	O_{jk}	Allocated Targets
	O_{11}	T_4, T_2, T_7
S_1	O_{12}	T_3, T_9
	O_{13}	T_{10}, T_{15}
	O_{21}	T_5, T_{13}, T_{15}
S_2	O_{22}	T_{23}
	O_{23}	T_{56}

4.1 Objective Evaluation

The approach for optimizing the scheduling of earth observation satellites was developed using Dijkstra's algorithm as an objective function. The objective was to maximize the total weighted score of the targets while respecting the memory and session time constraints of the satellite. weighted score is the function of priority and quality. It is calculated by using the equation 4.1.

$$\text{Weighted score } = \text{Priority } \times \text{Quality} \tag{4.1}$$

Dijkstra's algorithm is a well-known algorithm for finding the shortest path between nodes in a graph. But in this project, Dijkstra's is used as resource constraint longest path algorithm.

The satellite scheduling problem was modeled as a directed acyclic graph, where each node represented a time slot in the visibility window of a target, and each edge represented a transition between time slots that satisfied the imaging duration and transition time constraints. Dijkstra's algorithm was applied to this graph to find the longest path between first target and last target that maximized the total priority.

To apply Dijkstra's algorithm to the satellite scheduling problem, metrics such as target priorities, imaging duration, transition times, session time, and memory capacity were defined as inputs to the algorithm. The weight of the edges in the graph was calculated based on these metrics, such that higher priority targets had higher weights.

Dijkstra's algorithm was run on the graph, starting from the first target in the visibility window of the satellite and ending at the last target. The algorithm iteratively computed the path until it reached the last target. The algorithm was adapted to take into account the memory and session time constraints of the satellite, such that it would only consider paths that respected these constraints.

An example is presented to explain the objective evaluation procedure. The first solution in figure 4.1 is considered and decoded to obtain table 4.1, which displays the targets allocated for each satellite. For the first satellite, three targets - T2, T4, and T7 - are allocated. Target visibility to S1 is arranged as follows: T4, T2, T7. Additional information about the three targets and their possible transitions can be found in Figure 4.1. The operating time and memory used and weighted Score (4.1) up to a given node are updated using equation 4.2, equation 4.3 and equation 4.4 respectively.

$$\begin{pmatrix}\text{Operating time}\\ \text{used till}\\ \text{current node}\end{pmatrix} = \begin{pmatrix}\text{Operating time}\\ \text{used till}\\ \text{previous node}\end{pmatrix} + \begin{pmatrix}\text{previous}\\ \text{node's}\\ \text{Transition}\\ \text{time from}\end{pmatrix} + \begin{pmatrix}\text{current}\\ \text{node's}\\ \text{Imaging}\\ \text{duration}\end{pmatrix} \tag{4.2}$$

$$\begin{pmatrix}\text{Memory used}\\ \text{till current}\\ \text{node}\end{pmatrix} = \begin{pmatrix}\text{Memory used}\\ \text{till previous}\\ \text{node}\end{pmatrix} + \begin{pmatrix}\text{Memory}\\ \text{of current}\\ \text{node}\end{pmatrix} \tag{4.3}$$

$$\begin{pmatrix}\text{Weighted score}\\ \text{till current}\\ \text{node}\end{pmatrix} = \begin{pmatrix}\text{Weighted score}\\ \text{till previous}\\ \text{node}\end{pmatrix} + \begin{pmatrix}\text{Weighted}\\ \text{score of}\\ \text{current}\\ \text{node}\end{pmatrix} \tag{4.4}$$

To update the performance metrics of the satellite in the downloading stage, the operating time and weighted score are updated using the same equations as in the previous stages, as given in eq 4.2 and eq 4.4, respectively. However, the memory usage is updated differently in this stage. This is because the data captured by the satellite is transmitted to the ground station, resulting in an reducing the memory, making it ready to capture new images. Therefore, equation 4.5 is used for updating the memory during this stage.

$$\begin{pmatrix}\text{Memory avil-}\\ \text{able}\end{pmatrix} = \begin{pmatrix}\text{Memory used}\\ \text{till previous}\\ \text{node}\end{pmatrix} - \begin{pmatrix}\text{Memory}\\ \text{down-}\\ \text{loaded}\end{pmatrix} \tag{4.5}$$

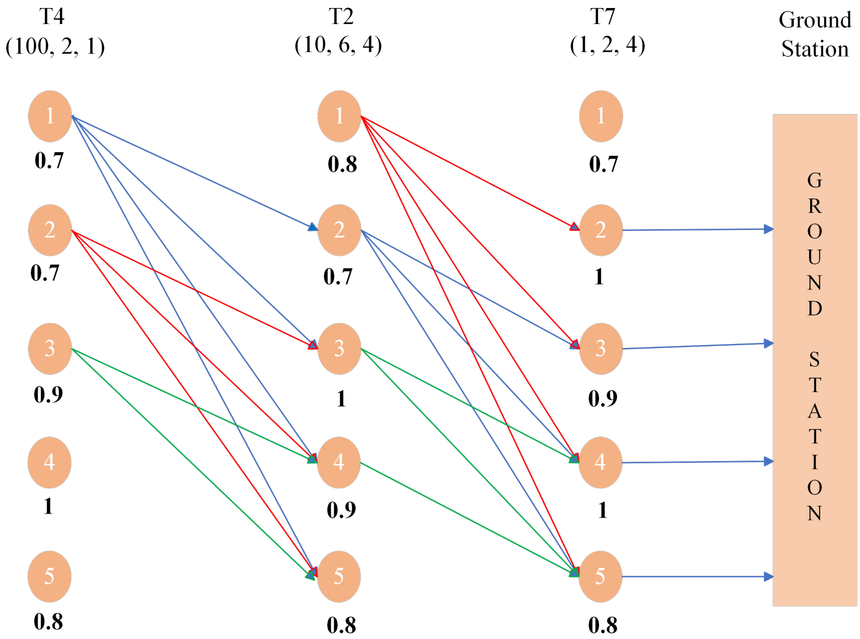

Fig. 4.1 Dijkstra Algorithm

In Example 4.1, each target is assigned a vector such as (100, 2, 1) for T_1. The first element of the vector represents the priority of the target, while the second element indicates the amount of memory storage that will be occupied if the target is captured. The third element represents the operating time to capture the target.

Each target is associated with a number of nodes, with each node having a corresponding quality value between 0.7 and 1. This quality value indicates the quality of the image that can be captured from that particular node. A value of 1 represents perfect image quality, while a value of 0.7 represents a lower quality image.

Stage-1:

Always choose the first target that is visible. Here T4 is the first visible target. The operation time for the T4 nodes is equal to the time needed for imaging, and the memory consumed is equal to the memory needed for T4 (see figure 4.1). Table 4.2 displays an overview of stage 1. There is no paths from nodes 4 and 5 of T4 so they are not include in table Table 4.2. In the second stage, nodes 1, 2, and 3 are extended to T2.

Stage-2:

Here discards nodes 1 of T2 because they don't have any incoming arcs. fig 4.1 shows the operating time, memory consumed, and weighted score achieved up to nodes 2, 3, 4 and 5 of

Table 4.2 Dijkstra Stage-1

Path no.	Nodes	Operating time	Memory	Weighted Score (Calculated by eq. 4.1)
1	1			70
2	2			70
3	3	1	2	90
4	4			100
5	5			80

Table 4.3 Dijkstra Stage-2

Path no.	Nodes		Operating Time	Memory	Weighted Score
	T4	T2	(Refer eq. 4.2)	(Refer eq. 4.3)	(Refer eq. 4.1 and 4.4)
1	1	2	$1+3+4=8$	$2+6=8$	$70+10\times0.7=77$
2	1	3	$1+4+4=9$	$2+6=8$	$70+10\times1=80$
3	1	4	$1+5+4=8$	$2+6=8$	$70+10\times0.9=79$
4	1	5	$1+6+4=11$	$2+6=8$	$70+10\times0.8=78$
5	2	3	$1+5+4=10$	$2+6=8$	$70+10\times1=80$
6	2	4	$1+6+4=11$	$2+6=8$	$70+10\times0.9=79$
7	2	5	$1+6+4=11$	$2+6=8$	$70+10\times0.8=78$
8	3	4	$1+5+4=10$	$2+6=8$	$90+10\times0.9=99$
9	3	4	$1+4+4=9$	$2+6=8$	$90+10\times0.8=98$

T2 and the paths formed up to this stage. Starting from node 1 of target T4, the satellite has the option to move to nodes 2, 3, 4, and 5 of target T2 via paths 1 to 2, 1 to 3, 1 to 4, and 1 to 5. Similarly, extending from node 2 of T2, the satellite can reach nodes 3, 4, and 5 of T2 via paths 2 to 3, 2 to 4, and 3 to 4. Lastly, from node 4 of T2, the satellite has the option to move to nodes 4 and 5 of T2. These paths are extended to nodes of T7 in the third stage.

All the paths shown in Table 4.3 satisfy the constraints on time and memory. If the evaluation is at the final step without considering the third target T7, the algorithm selects one of the paths with the highest weighted score, which is path no. 8.

Stage-3:

Node-1 is discarded because there are no incoming arcs. The algorithm can start from node-1 of the target T2 and extend to node-2, 3, 4, and 5 of target T7. Path-1 can be extended (see Table 4.3) to nodes-3, 4, and 5 of target T7, with available paths of (1-2-3, 1-2-4, and 1-2-5). Similarly, path-2 (see Table 4.3) can be extended to nodes-4 and 5 of target T7, with available paths of (1-3-4 and 1-3-5). Table 4.4 displays an overview of stage-3.

Table 4.4 Dijkstra Stage-3

Path no.	Nodes			Operating Time	Memory	Weighted Score
	T4	T2	T7	(Refer eq. 4.2)	(Refer 4.3)	(Refer eq. 4.1 and eq. 4.4)
1	-	1	2	$4+5+4=13$	$6+2=8$	$10\times0.8+1\times1=9$
2	-	1	3	$4+4+4=12$	$6+2=8$	$10\times0.8+1\times0.9=8.9$
3	-	1	4	$4+4+4=12$	$6+2=8$	$10\times0.8+1\times1=9$
4	-	1	5	$4+3+4=11$	$6+2=8$	$10\times0.8+1\times0.8=8.8$
5	1	2	3	$8+4+4=16$	$2+6+2=10$	$77+1\times0.9=77.9$
6	1	3	4	$9+5+4=18$	$2+6+2=10$	$80+1\times1=81$
7	1	4	5	$8+3+4=15$	$2+6+2=10$	$79+1\times0.8=79.8$
8	2	3	4	$10+5+4=19$	$2+6+2=10$	$80+1\times1=81$
9	2	3	5	$10+5+4=19$	$2+6+2=10$	$80+1\times0.8=80.8$
9	2	4	5	$11+3+4=18$	$2+6+2=10$	$79+1\times0.8=79.8$
10	3	4	5	$10+3+4=17$	$2+6+2=10$	$99+1\times0.8=99.8$

Stage-4:

In this stage, which is referred to as the Downloading Stage, all data from the previous stages are downloaded to the ground station. It is assumed that if the connection between the satellite and ground station is established for more than 100 seconds, then all data are downloaded successfully.

It's important to note that every path, including those shown in tables 4.2, 4.3, and 4.4, must pass through this stage in order for the satellite to be ready for further movement. During this stage, the data from the previous stages are transmitted to the ground station. Table 4.5 displays an overview of stage-4.

After completing the four steps, it can be observed that path-10 (as referred in Table 4.5) has the highest weighted score among all the paths. Hence, the optimal path is determined to be 3-4-5, which means moving from node-3 of target T4 to node-4 of target T2 and finally reaching node-5 of target T7.

4.2 Genetic Algorithm

Genetic Algorithms (GAs) belong to the family of evolutionary algorithms and are widely used for solving optimization and search problems. They are adaptive heuristic search algorithms that mimic the natural selection process by which species that can adapt to their environment survive and reproduce. This selection process is commonly known as "survival of the fittest."

Table 4.5 Dijkstra Stage-4

Path no.	Nodes			Operating Time	Memory	Weighted Score
	T4	T2	T7	(Refer eq. 4.2)	(Refer 4.5)	(Refer eq. 4.1 and eq. 4.4)
1	-	1	2	$13+5=17$	$8-8=0$	$10\times0.8+1\times1=9$
2	-	1	3	$12+4=16$	$8-8=0$	$10\times0.8+1\times0.9=8.9$
3	-	1	4	$12+5=17$	$8-8=0$	$10\times0.8+1\times1=9$
4	-	1	5	$11+4=15$	$8-8=0$	$10\times0.8+1\times0.8=8.8$
5	1	2	3	$16+4=20$	$10-10=0$	$77+1\times0.9=77.9$
6	1	3	4	$18+3=21$	$10-10=0$	$80+1\times1=81$
7	1	4	5	$15+4=19$	$10-10=0$	$79+1\times0.8=79.8$
8	2	3	4	$19+5=24$	$10-10=0$	$80+1\times1=81$
9	2	3	5	$19+4=23$	$10-10=0$	$80+1\times0.8=80.8$
9	2	4	5	$18+6=24$	$10-10=0$	$79+1\times0.8=79.8$
10	3	4	5	$17+7=23$	$10-10=0$	$99+1\times0.8=99.8$

GAs are based on the idea of simulating genetic behavior and structure of chromosomes in a population. In a GA, each individual in the population represents a possible solution in the search space and is typically encoded as a string of characters, integers, floats, or bits, similar to a chromosome in biology. The genes within the chromosome correspond to the variable components of the solution.

Individuals within a GA population compete for resources and can mate to create offspring. The fittest individuals, as determined by their fitness function, have a higher probability of mating and propagating their genes to the next generation. This creates new individuals that combine genes from both parents and may lead to offspring that are better suited to the environment than their parents. In this way, GA populations evolve towards better solutions over successive generations.

4.2.1 Fitness Score

In a genetic algorithm, each individual in the population is assigned a fitness score that represents its ability to compete in the environment. The algorithm maintains a population of n individuals, with each individual representing a possible solution to the problem at hand. The individuals with better fitness scores have a higher chance of reproducing and passing on their genes to the next generation.

When all available options for mating with the previous population have been used up, a new generation will emerge because the population number will continue to be stagnant. Over successive generations, the algorithm aims to generate better solutions by allowing the fittest individuals to reproduce and propagate their genes.

Each new generation typically has better genes than the previous one, which leads to better partial solutions. Once there are no discernible differences between the offspring generated and those produced by prior populations, the algorithm converges and resulting in a set of solutions for the problem.

4.2.2 Operators of Genetic Algorithms

The genetic algorithm creates the initial generation and then uses three major operators to evolve the population:
1. Selection Operator 2. Crossover Operator 3. Mutation Operator
The GA is executed using the flowchart presented in Figure 4.2.

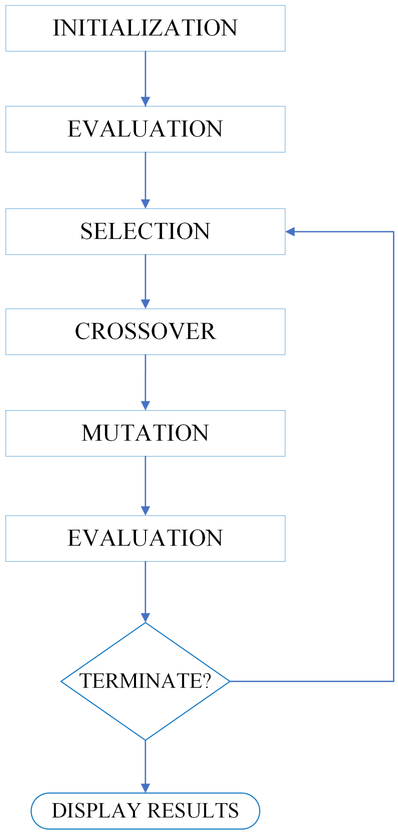

Fig. 4.2 Genetic Algorithm Flow Chart

Selection Operator:

This operator gives preference to individuals with good fitness scores, allowing them to pass their genes to the next generation. There are various types of selection operators, including tournament selection, roulette wheel selection, and rank-based selection, which choose individuals for reproduction based on their fitness score. The roulette wheel selection process is executed using the flowchart presented in Figure 4.3.

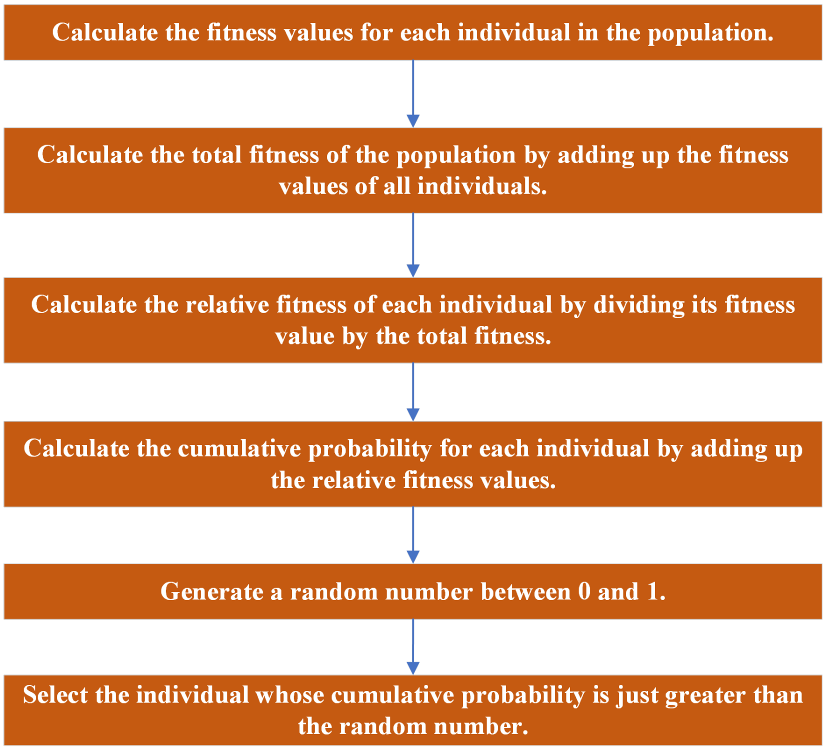

Fig. 4.3 Flowchart for Roulette Wheel Selection

Crossover Operator:

This operator represents mating between different individuals. Using the selection operator, two individuals are chosen, and their genes are then randomly swapped at crossover points to form a new individual. The crossover operation is executed using the flowchart presented in Figure 4.3. Population before crossover and after crossover is shown in the fig 4.4.

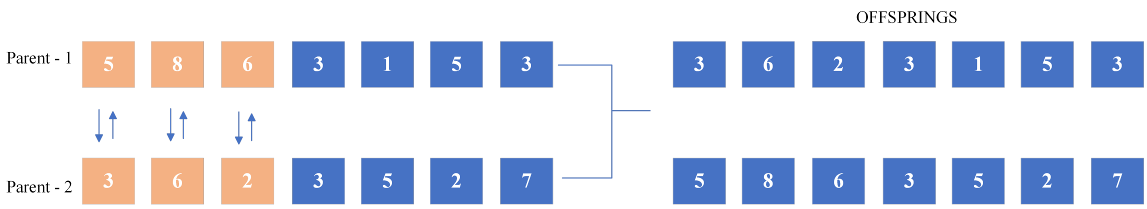

Fig. 4.4 Graphical Representation of Crossover

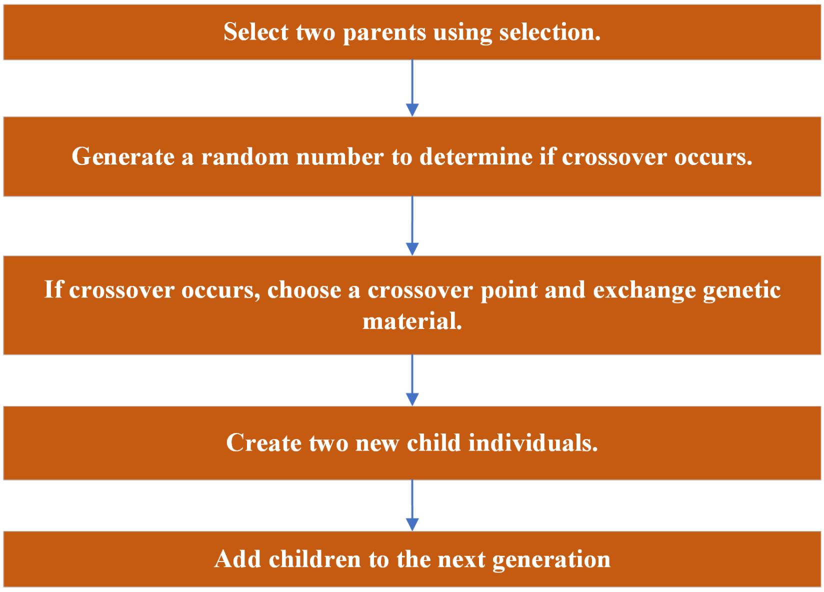

Fig. 4.5 Flowchart for Crossover Operation

Mutation Operator:

This operator introduces random changes in offspring to maintain population diversity and avoid premature convergence. The mutation process is executed using the flowchart presented in Figure 4.7. Fig 4.6 represent the individual before mutation and after mutation.

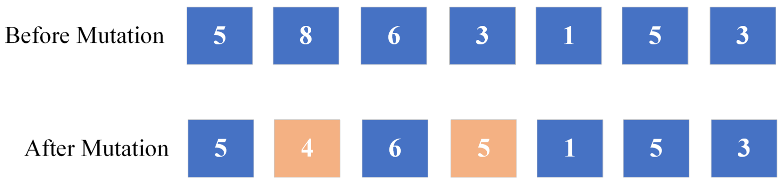

Fig. 4.6 Graphical Representation of Mutation

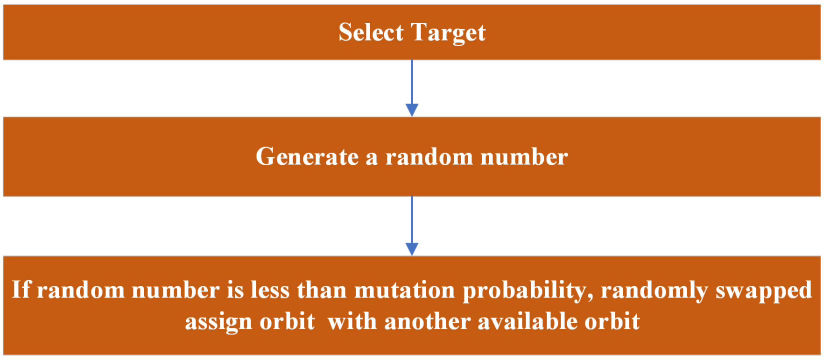

Fig. 4.7 Flowchart for Mutation Operation

4.2.3 Proposed Algorithm

The first step in the proposed algorithm is to generate an initial population of candidate solutions for the scheduling problem. Each solution in the population represents a possible scheduling plan for the satellite, with the genes encoding the selected satellite orbit for that particular target. In fig 4.8, the number inside the square box is index representing allocated satellite orbit for each target or ground station.

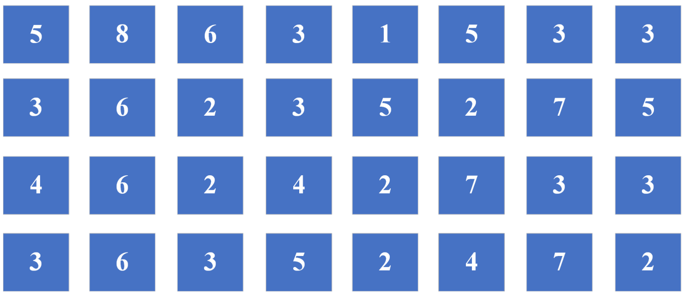

Fig. 4.8 Example of Initial Population

Once the initial population has been generated (fig 4.8), the fitness of each solution is evaluated using a profit function. In this case, the Dijkstra Algorithm is used as the profit function. This profit function represents the overall profit that the solution can generate for the satellite. The fitness value of each solution is calculated based on its profit function score. Profit score for each option is present in the Table 4.6.

After evaluating the fitness of all solutions, the top-performing individuals are reserved using an elitism process. This is done to preserve the best solutions and prevent them from being lost in the subsequent crossover and mutation steps. The top three solutions are reserved during the Elitism process, as shown in Table 4.7 where they are highlighted in green.

Table 4.6 Profit Score

Initial Population								Profit
5	8	6	3	1	5	3	3	943
3	6	2	3	5	2	7	5	1040
4	6	2	4	2	7	3	3	1057
3	6	3	5	2	4	7	2	990
1	9	3	8	2	7	1	3	1145
5	6	3	9	2	10	3	8	1074
6	4	4	3	1	4	10	2	1235
3	2	5	1	9	8	5	3	1075
8	9	7	7	6	1	7	1	1054
7	1	9	4	4	6	3	6	1134

Table 4.7 Elite Process

Initial Population								Profit
5	8	6	3	1	5	3	3	943
3	6	2	3	5	2	7	5	1040
4	6	2	4	2	7	3	3	1057
3	6	3	5	2	4	7	2	990
1	9	3	8	2	7	1	3	1145
5	6	3	9	2	10	3	8	1074
6	4	4	3	1	4	10	2	1235
3	2	5	1	9	8	5	3	1075
8	9	7	7	6	1	7	1	1054
7	1	9	4	4	6	3	6	1134

Table 4.8 Roulette wheel

Initial Population								Profit	Probability	Cumulative Probability	Random Number	New Population						
5	8	6	3	1	5	3	3	943	0.130	0.130	0.488	3	6	3	5	2	4	7
3	6	2	3	5	2	7	5	1040	0.144	0.274	0.257	4	6	2	4	2	7	3
4	6	2	4	2	7	3	3	1057	0.146	0.420	0.566	5	6	3	9	2	10	3
3	6	3	5	2	4	7	2	990	0.137	0.557	0.746	3	2	5	1	9	8	5
5	6	3	9	2	10	3	8	1074	0.148	0.706	0.824	3	2	5	1	9	8	5
3	2	5	1	9	8	5	3	1075	0.149	0.854	0.922	8	9	7	7	6	1	7
8	9	7	7	6	1	7	1	1054	0.146	1.000	0.090	3	2	5	1	9	8	5

Table 4.9 Population After Crossover

Population								Crossover Probability (0.8)	New Population							
3	6	3	5	2	4	7	2	Y	4	6	2	5	2	4	7	2
4	6	2	4	2	7	3	3	Y	3	6	3	4	2	7	3	3
5	6	3	9	2	10	3	8	Y	3	2	5	1	2	10	3	8
3	2	5	1	9	8	5	3	N	3	2	5	1	9	8	5	3
3	2	5	1	9	8	5	3	Y	5	6	3	9	9	8	5	3
8	9	7	7	6	1	7	1	Y	3	2	7	7	6	1	7	1
3	2	5	1	9	8	5	3	Y	8	9	5	1	9	8	5	3

After the elite individuals have been reserved, the remaining solutions in the population are selected as parents for the next generation using Roulette wheel selection. Roulette wheel process is used by following the flowchart 4.3. The resulting population after applying the roulette wheel selection is presented in Table 4.8.

The next step is crossover, in which two solutions are randomly selected from the population and combined to create new offspring solutions. Cross over is presented in Fig 4.4. To perform crossover, a probability of 0.8 is used. A random number between 0 and 1 is generated, and if it is less than 0.8, crossover is performed. If an odd number of options are available for crossover, one is left out and the genes are paired from the left for crossover. A random number between 0 and the length of the gene is generated for the crossover point. The resulting population after applying the crossover is presented in Table 4.9.

Next step is mutation. In the mutation step, some of the genes in the population are randomly altered, introducing further variation into the solutions. This helps prevent the population from becoming stuck in a local optimum and encourages exploration of the solution space. Example of mutation is shown in Fig 4.10.

Table 4.10 Population After Mutation

Population								Population after Mutation							
4	6	2	5	2	4	7	2	4	4	2	5	3	4	7	2
3	6	3	4	2	7	3	3	3	6	5	4	2	5	3	3
3	2	5	1	2	10	3	8	3	4	5	1	2	10	3	8
3	2	5	1	9	8	5	3	3	2	5	3	9	8	5	3
5	6	3	9	9	8	5	3	5	2	3	9	9	4	5	3
3	2	7	7	6	1	7	1	3	2	7	7	6	1	7	1
8	9	5	1	9	8	5	3	8	9	3	1	9	8	5	3

Evaluating the Fitness of the New Population: After the crossover and mutation steps, the fitness of the new population is evaluated using the profit function (Dijkstra Algorithm). This step determines the profit potential of the new solutions and allows them to be compared to the previous population.

Then, the elite individuals that were reserved in step 3 are reintroduced into the population, maintaining the best solutions found so far and preventing the algorithm from losing progress.

Repeat this for a specified number of generations or until a certain termination criterion is met. The termination criterion could be reaching a specific fitness value or a maximum number of generations. At the end of the algorithm, the best solution found in the population is selected as the final solution for satellite scheduling. The pseudo code for genetic algorithm is shown in the fig 4.9.

4.3 Population Based Iterated Local Search

Iterated Local Search (ILS) is a powerful metaheuristic optimization method that has proven effective in solving combinatorial optimization problems. Its approach involves iteratively generating candidate solutions within the solution space and then refining them using a local search algorithm.

In the ILS framework, an initial solution is randomly generated, and a local search algorithm is employed to improve its quality. This iterative process of improvement continues by perturbing the improved solution to generate a new candidate solution, which is further refined through the local search algorithm. The iterations persist for a predetermined number of steps or until a satisfactory solution is achieved.

The fundamental concept behind ILS lies in the recognition that by perturbing a previously good solution and re-optimizing it using local search, a better solution can often be obtained.

Fig. 4.9 Pseudocode of Genetic Algorithm [2]

Require: A fitness function f and a set of parameters, such as population size N, crossover probability p_c, and mutation probability p_m.
Ensure: A solution that maximizes the fitness function f.

Initialize an initial population P_0 of N random solutions.
Set the generation count g to zero.
repeat
  Evaluate the fitness of each solution in population P_{gen} using the function f. Create an empty population P_{gen+1}.
  Copy the fittest solution from population P_{gen} to P_{gen+1} (elitism).
  repeat
    Apply selection operator(Roulette wheeel), select two parent solutions from population P_{gen}
    Apply the crossover with probability p_c to produce offspring solutions.
    Apply the mutation operator with probability p_m, to the offspring solutions.
    Add the resulting solutions to the new population P_{g+1}.
  until the size of population P_{g+1} reaches N.
  Set the generation count g to $g+1$.
until a termination criterion is met.
Return the solution with the highest fitness value from population P_g.

Population-based Iterated Local Search (PBILS) follows a similar methodology to ILS. However, PBILS begins with an initial population of candidate solutions and proceeds to iteratively improve these solutions through a series of local search operations and population updates. PBILS is particularly advantageous for combinatorial optimization problems due to its ability to explore a larger search space. Unlike regular local search algorithms, PBILS can generate a diverse set of candidate solutions, avoiding the trap of local optima and producing higher-quality solutions. Additionally, PBILS lends itself well to parallelization, which can greatly accelerate the optimization process for large-scale problems. Figure 4.10 provides a visual representation of the flow chart for Population Based Iterated Local Search. And figure 4.11 showcases the pseudo code for the Population-based Iterated Local Search algorithm.
To elaborate further, let us describe the specific steps involved in Population-based Iterated Local Search:

Step 1: Generation of Initial Population:
The first step in PBIL is to generate an initial population, which consists of a set of candidate solutions. This population can be randomly generated, ensuring a diverse set of potential solutions.

Step 2: Fitness Evaluation:
Once the initial population is generated, the next step involves evaluating the fitness value of each candidate solution using a developed objective function. In this case, we employ Dijkstra's algorithm to assess the fitness of each candidate solution.

Step 3: Neighborhood Exploration:
In this step, the algorithm explores the neighborhood of the initial population by considering reassignment of a target to a different satellite orbit. A probability factor is introduced to determine whether a reassignment operation should be performed. If the probability factor is less than a predefined threshold probability, the reassignment operation is carried out. Otherwise, it is skipped. This probabilistic approach allows for exploration and potential improvement of the solutions.

Step 4: Fitness Score Evaluation:
After the neighborhood exploration, the fitness score of the new population is evaluated. If the fitness score of the new population is found to be greater than the fitness score of the previous population, it indicates an improvement. In such a case, the algorithm proceeds to the next iteration using the new population. However, if the fitness score does not improve, the

algorithm continues with the old population. This iterative process is repeated for a specific number of iterations, enabling the algorithm to converge towards optimal or near-optimal solutions.

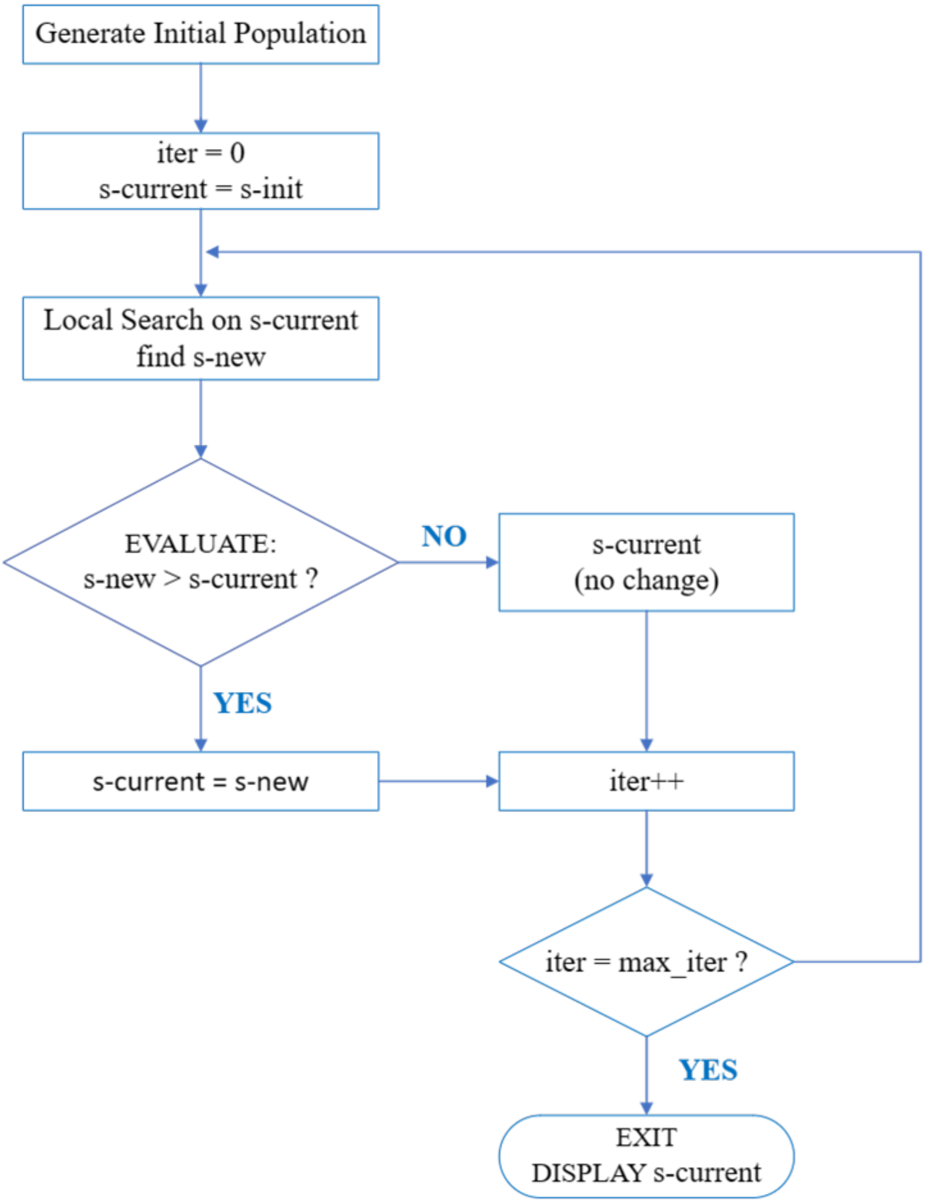

Fig. 4.10 Iterated Local Search Flow Chart

Step-5: Termination Criteria In this step, a termination condition is established to determine when to halt the PBILS algorithm. This condition may involve factors such as a specified number of iterations, achieving a desired fitness threshold, or reaching a predefined computational time limit. The termination condition ensures the algorithm stops when a satisfactory solution is found or when further iterations are unlikely to yield significant improvements. The best solution obtained throughout the iterations is considered the final output.

Fig. 4.11 Population Based Iterated Local Search [3]

```
Require: fitness function (f), probability factor p_c
Ensure: obj_w(gbest)
Generate Initial Population (|X|)
for x_i ∈ X do
    obj_w(x_i) ← Objective(x_i)
    x_i ← Local Search
    if obj_w(gbest) < obj_w(x_i) then
        gbest ← x_i
    end if
end for
while maximum iterations not reached do
    for x_i ∈ X do
        obj_w(x_i) ← Priority Objective(x_i)
        x_i ← Local Search
        if obj_w(gbest) < obj_w(x_i) then
            gbest ← x_i
        end if
    end for
end while
return obj_w(gbest)
```

Chapter 5

Results

5.1 Test Problem

7 test problem are consider for each of the proposed algorithms The number of targets in the test problems ranged from 30 to 150, and the number of satellite range from 2 to 5. The roll and pitch limits of the satellites used in the experiments are shown in the table 5.1 along with their other characteristics. The scheduling approach assumes values for session time, on-board memory size, and angular velocity. These values are used to guide the scheduling optimization process. The data used in this process is generate using Ansys STK mission design software.

The test problems were given the names Sx-Ty, where x stands for the number of targets and y for the number of satellites. The table's roll and pitch limits precisely reflected the characteristics of the satellites used in the experiments.

These test problems were used to assess the proposed algorithms, and their performance was examined in terms of a number of factors, including computation time and solution quality. The effectiveness of each algorithm was then evaluated, along with potential areas for improvement, using the results of the experiments.

Overall, the test problems provided a rigorous evaluation of the proposed algorithms and enabled a thorough analysis of their performance. The use of a range of test problems with varying numbers of targets and satellites helped to ensure that the algorithms were evaluated under realistic and diverse conditions, which is crucial for determining their effectiveness in real-world scenarios.

Table 5.1 Available Satellite and it's properties [4, 5]

Satellite Name	Roll		Pitch		E	M	ω
	ϕ^+ (deg)	ϕ^- (deg)	θ^+ (deg)	θ^- (deg)	(sec)	(MB)	(deg/sec)
CARTOSAT 1	18	-18	26	-10	600	150	0.25
CARTOSAT 2	17.5	-17.5	26	-10	600	150	0.25
CARTOSAT 2A	17.5	-17.5	26	-10	600	150	0.25
CARTOSAT 2B	17.5	-17.5	26	-10	600	150	0.25
CARTOSAT 2C	21.5	-21.5	26	-10	600	150	0.25
CARTOSAT 21)	21.5	-21.5	26	-10	600	150	0.25
CARTOSAT 2E	21.5	-21.5	26	-10	600	150	0.25
CARTOSAT 2F	21.5	-21.5	26	-10	600	150	0.25
CARTOSAT 3	21.5	-21.5	26	-10	600	150	0.25
RESOURCESAT 1	25.5	-25.5	3.5	-3.5	550	180	0.5
RESOURCESAT 2	24.5	-24.5	3.5	-3.5	550	180	0.5
RESOURCESAT 2A	24.5	-24.5	3.5	-3.5	550	180	0.5
MEGHA-TROPIQUES	62.5	-62.5	4	-4	700	200	0.75
OCEANSAT 2	44.5	-44.5	2	-2	700	200	0.75
HYSIS	1.5	-1.5	3	-3	500	100	1

5.2 Parameter estimation and Sensitivity analysis

To evaluate how the algorithm's performance is affected by variations in the population size and number of iterations, two experiments were carried out. The results of these experiments are discussed in this section, which analyzes the sensitivity of the algorithm's performance.

5.2.1 Population Variation Analysis

The population size was varied from 40 to 120 individuals with increments of 40 individuals. The algorithm was run with a fixed number of iterations. The results were analyzed in terms of the mean objective function value and computation time. The results of the population variation analysis for Genetic Algorithm and Population Based Iterative Local search are presented in Tables 5.2 and 5.3, respectively. These tables provide insights into the performance of each algorithm under varying population sizes.

By examining the data presented in these tables, it is possible to determine the population size that yields the best results for each algorithm, as well as the point of diminishing returns where increasing the population size no longer leads to a significant improvement in performance. Overall, this analysis provides valuable information for selecting appropriate population sizes.

Table 5.2 Variation with population size for GA

Test Problem		Population Size 40	80	120
S4-T50	Objective	907	907	908
	Computation Time (min)	85	133	179
S5-T100	Objective	1909	1910	1998
	Computation Time (min)	125	209	357
S5-T150	Objective	2225	2354	2245
	Computation Time (min)	204	321	218

Table 5.3 Variation with population size for PBIL

Test Problem		Population Size 5	10	20
S4-T50	Objective	800	797	907
	Computation Time (min)	30	85	313
S5-T100	Objective	2006	1998	2008
	Computation Time (min)	55	142	405
S5-T150	Objective	2495	2487	2485
	Computation Time (min)	73	273	490

5.2.2 Iteration Variation Analysis

In order to investigate the impact of the number of iterations on the algorithm's performance, a series of experiments were carried out by varying the number of iterations from 4 to 100 for PBIL and 4 to 40 for GA with an increment of 4. It is important to note that the population size remained fixed throughout the experiments. The results of the analysis on the effect of iteration variation on the performance of PBIL and GA are presented in Figure 5.2 and 5.1 respectively.

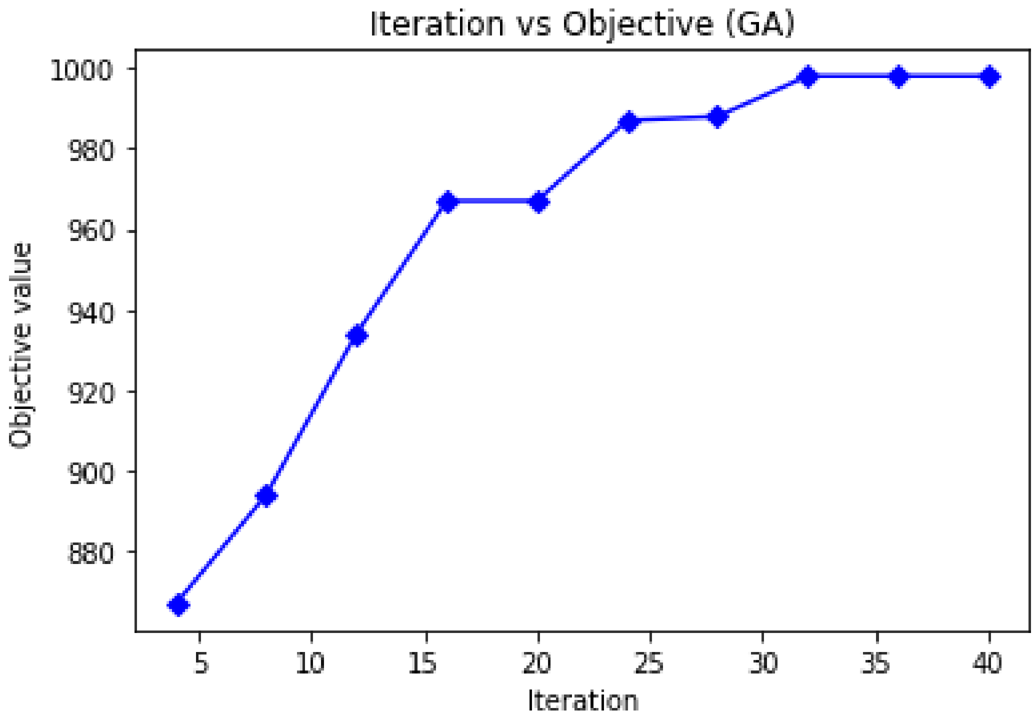

Fig. 5.1 Variation of Objective value as number of Iteration Increase

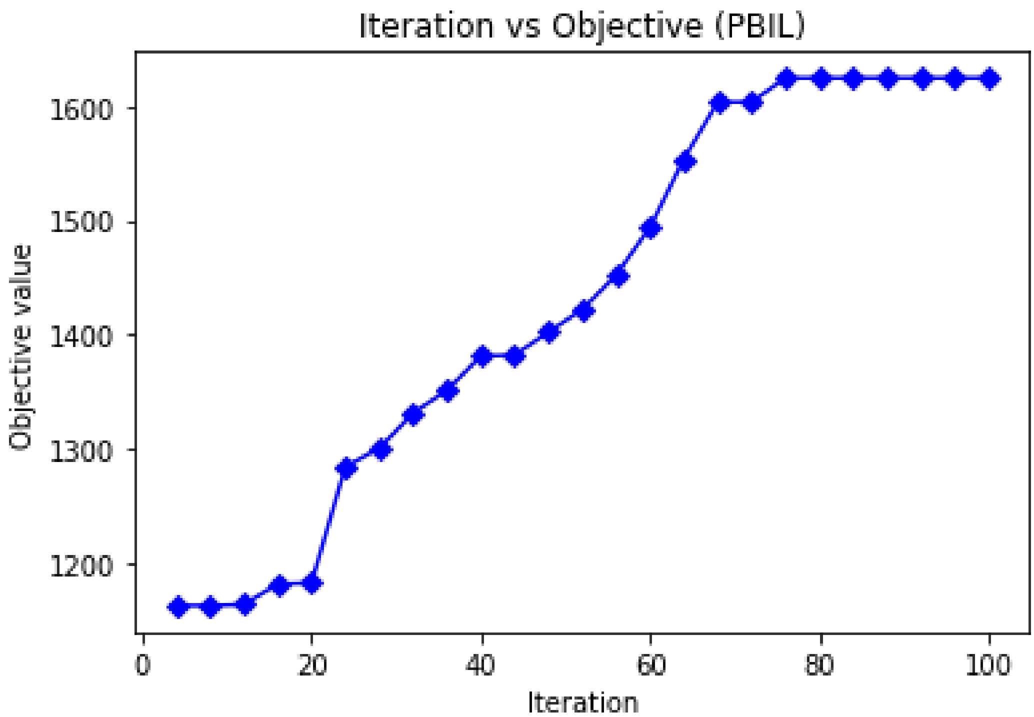

Fig. 5.2 Variation of Objective value as number of Iteration Increase

Table 5.4 Parameters Consider in GA

Population Size	40
Mutation Probability	0.05
Crossover probability	0.85

Table 5.5 Parameters Consider in PBIL

Population Size	10
Probability factor	0.4

As can be seen from the graph in fig 5.1 and fig 5.2, the objective value initially improves as the optimization process proceeds, but eventually reaches a point where there is little to no further improvement despite additional iterations of the algorithm. At the beginning of the search, the algorithm is still exploring the search space and is likely to find better solutions, which leads to an increase in the objective function value. However, as the search progresses, the algorithm gradually converges towards a solution that is close to the optimal solution. At this point, the algorithm has explored most of the promising regions of the search space, and further improvement becomes difficult. Therefore, the objective function value becomes constant as the algorithm has reached a point where it can no longer find better solutions.

5.3 Performance comparison

This section presents a comparison of the performance of the Genetic Algorithm (GA) and Population-Based Iterated Local Search (PBIL), Impertialist Competitive Algorithm and Artificial Bee Colony Algorithm. The comparison is based on the objective values achieved by each algorithm and the computation time required to obtain the solutions. Table 5.6 summarizes the comparison results and provides insights into the strengths and weaknesses of each algorithm. The parameters consider in GA is shown in the tab 5.4 and parameters for PBIL is shown in table 5.5.

From the table 5.6, it can be observed that ICA and ABC algorithm has a lower computation time compared to proposed algorithm. This is because of different in objective function (i.e. Dijkstra's algorithm). Two type of approach is used for evaluating objective function. In ICA and ABC both satellite orbit and slots are assigns to each target in initial state before applying Dijkstra's Algorithms. In contrast, in GA and PBIL case only the satellite orbit assign to target in initial state and handle slot assignment in Dijkstra's algorithm. Since there are numerous slots available for each target, the Dijkstra algorithm alone takes approximately 2 minutes for one time computation.

Table 5.6 Comparision between GA, PBIL, ICA and ABC

Test Problem		Fitness Function - 1 GA	PBIL	Fitness Function - 2 ICA	ABC
S2-T30	Average Objective	19	19	19.07	19.06
	Best Objective	19	19	19.09	19.09
	Computation Time (min)	0.5	0.4	0.334	9.198
S3-T50	Average Objective	200.5	201	217.038	220.668
	Best Objective	203	204	220.847	220.834
	Computation Time (min)	3.6	11.2	1.027	25.489
S4-T50(P1)	Average Objective	973	981	1238.39	1238.34
	Best Objective	1172	1177	1238.64	1238.64
	Computation Time (min)	12.5	46.7	26.4	34.15
S4-T50	Average objective	907.3	805.3	1023.39	1023.39
	Best objective	908	907	1021.76	1023.324
	Computation lime (min)	116	227	3.336	34.148
S8-Tl50	Average objective	3995.3	3998.7	4140.12	4032.98
	Best Objective	3998	4002	4083.5	3956.514
	Computation Tune (min)	230	405	4.755	5.504

Chapter 6

Conclusions and future scope

In conclusion, this report has conducted an analysis of two optimization algorithms, namely genetic algorithms (GA) and population-based iterated local search (PBILS), for the integrated scheduling of Earth Observation Satellites (EOS) and ground stations. The objective of this study was to evaluate the effectiveness and performance of these algorithms in solving the scheduling problem while considering various resource constraints such as transition time, energy, memory, and priority.

The proposed scheduling algorithm, utilizing both the genetic algorithm and population-based iterative local search, has yielded promising results. The algorithm successfully addressed the resource constraints through the implementation of a resource constraint Dijkstra algorithm, which provided a comprehensive evaluation of the objectives. By considering factors such as transition time, energy consumption, memory utilization, and task priority, the algorithm was able to optimize the scheduling process effectively.

The comparison between the GA and PBILS algorithms has provided valuable insights into their individual strengths and capabilities for scheduling tasks. The genetic algorithm demonstrated its ability to explore a wide search space and find globally optimal solutions. Its incorporation of genetic operators such as selection, crossover, and mutation allowed for efficient exploration and exploitation of the search space. On the other hand, the population-based iterated local search algorithm showcased its effectiveness in exploiting local search strategies and refining solutions iteratively. This made it particularly suitable for fine-tuning and improving the quality of solutions obtained through the initial genetic algorithm.

Future Scope:

The findings of this study have laid a solid foundation for future research and improvements in the field of integrated scheduling of Earth Observation Satellites (EOS) and ground stations.

As the project concludes, it is important to highlight potential areas of future work that can further enhance the effectiveness and efficiency of the developed scheduling algorithm.

1. Optimizing Objective Function Evaluation: One crucial aspect to address in future work is optimizing the objective function evaluation process. Although the proposed algorithm has shown promising results, the evaluation time currently takes approximately 2 minutes.

2. Incorporating Complex Constraints: While the present study considered important resource constraints such as transition time, energy, memory, and priority, there is room for further improvement by incorporating more complex constraints. Future work could explore the inclusion of constraints related to weather conditions or satellite subsystem failures.

3. Multiple Quality Metrics: The current algorithm evaluates a single quality metric ranging from 0.7 to 1. However, future work could explore the incorporation of additional quality metrics to enhance scheduling decisions. For instance, including metrics such as image resolution or data accuracy can enable the algorithm to prioritize tasks that require higher quality output.

References

[1] F. Yao, "Modeling and solving for multi-satellite cooperative task allocation problem based on genetic programming method." [Online]. Available: https://www.researchgate.net/scientific-contributions/Feng-Yao-2132022106

[2] "Efficient elicitation of software configurations using crowd preferences and domain knowledge." [Online]. Available: https://link.springer.com/article/10.1007/s10515-018-0247-4

[3] H. Shah-Hosseini, "Multilevel thresholding for image segmentation using the galaxy-based search algorithm." [Online]. Available: http://dx.doi.org/10.5815/ijisa.2013.11.03

[4] W. contributors, "Landsat program." [Online]. Available: https://en.wikipedia.org/wiki/Landsat_program

[5] S. B. R.K. Jaiswal, "Indian earth observation program," 2018. [Online]. Available: https://doi.org/10.1016/B978-0-12-409548-9.10321-5

[6] M. Lemaıtre, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, 2002. [Online]. Available: https://doi.org/10.1016/S1270-9638(02)01173-2

[7] M. Lemaître, G. Verfaillie, F. Jouhaud, J. Lachiver, and N. Bataille, "How to manage the new generation of agile earth observation satellites." Citeseer, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:11005745

[8] C. Mancel and P. Lopez, "Complex optimization problems in space systems," in *13Th International Conference on Automated Planning & Scheduling (ICAPS'03)*, 2003. [Online]. Available: https://hal.science/hal-00138203/

[9] F. Bunkheila, E. Ortore, and C. Circi, “A new algorithm for agile satellite-based acquisition operations,” *Acta Astronautica*, vol. 123, pp. 121–128, 2016. [Online]. Available: https://doi.org/10.1016/j.actaastro.2016.03.023

[10] P. Xie, H. Wang, Y. Chen, and P. Wang, “A heuristic algorithm based on temporal conflict network for agile earth observing satellite scheduling problem,” *IEEE Access*, vol. 7, pp. 61 024–61 033, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2902669

[11] X. Chen, G. Reinelt, G. Dai, and M. Wang, “Priority-based and conflict-avoidance heuristics for multi-satellite scheduling,” *Applied Soft Computing*, vol. 69, pp. 177–191, 2018. [Online]. Available: https://doi.org/10.1016/j.asoc.2018.04.021

[12] D. Habet, M. Vasquez, and Y. Vimont, “Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite,” *Computational optimization and applications*, vol. 47, pp. 307–333, 2010. [Online]. Available: https://doi.org/10.1007/s10589-008-9220-7

[13] Y. Xu, X. Liu, R. He, and Y. Chen, “Multi-satellite scheduling framework and algorithm for very large area observation,” *Acta Astronautica*, vol. 167, pp. 93–107, 2020. [Online]. Available: https://doi.org/10.1016/j.actaastro.2019.10.041

[14] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, “Daily imaging scheduling of an earth observation satellite,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 2, pp. 213–223, 2005. [Online]. Available: https://doi.org/10.1109/TSMCA.2005.843380

[15] E. Bensana, M. Lemaitre, and G. Verfaillie, “Earth observation satellite management,” *Constraints*, 1999. [Online]. Available: https://doi.org/10.1023/A:1026488509554

[16] W. J. Wolfe and S. E. Sorensen, “Three scheduling algorithms applied to the earth observing systems domain,” *Management Science*, vol. 46, no. 1, pp. 148–166, 2000. [Online]. Available: https://doi.org/10.1287/mnsc.46.1.148.15134

[17] V. Gabrel and D. Vanderpooten, “Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite,” 2002. [Online]. Available: https://doi.org/10.1016/S0377-2217(01)00188-6

[18] V. Gabrel and C. Murat, “Mathematical programming for earth observation satellite mission planning,” *Operations research in space and air*, pp. 103–122, 2003. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4757-3752-3_7

[19] N. Bianchessi, J.-F. Cordeau, J. Desrosiers, G. Laporte, and V. Raymond, “A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites,” *European Journal of Operational Research*, vol. 177, no. 2, pp. 750–762, 2007. [Online]. Available: https://doi.org/10.1016/j.ejor.2005.12.026

[20] N. Bianchessi and G. Righini, “Planning and scheduling algorithms for the cosmo-skymed constellation,” *Aerospace Science and Technology*, vol. 12, no. 7, pp. 535–544, 2008. [Online]. Available: https://doi.org/10.1016/j.ast.2008.01.001

[21] P. Wang and G. Reinelt, “A heuristic for an earth observing satellite constellation scheduling problem with download considerations,” *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 711–718, 2010. [Online]. Available: https://doi.org/10.1016/j.endm.2010.05.090

[22] K. T. Malladi, S. M. Minic, D. Karapetyan, and A. P. Punnen, “Satellite constellation image acquisition problem: A case study,” *Space Engineering: Modeling and Optimization with Case Studies*, pp. 177–197, 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-41508-6_7

[23] M. Barkaoui and J. Berger, “A new hybrid genetic algorithm for the collection scheduling problem for a satellite constellation,” *Journal of the Operational Research Society*, vol. 71, no. 9, pp. 1390–1410, 2020. [Online]. Available: https://doi.org/10.1080/01605682.2019.1609891

[24] D.-H. Cho, J.-H. Kim, H.-L. Choi, and J. Ahn, “Optimization-based scheduling method for agile earth-observing satellite constellation,” *Journal of Aerospace Information Systems*, vol. 15, no. 11, pp. 611–626, 2018. [Online]. Available: http://doi.org/10.2514/1.I010620

[25] S. V. Janakiram, “Optimal scheduling strategies for multiple agile earth observation satellites,” Indian Institute of Space Science and Technology, B.Tech project report, 2020.

[26] G. Wu, H. Wang, W. Pedrycz, H. Li, and L. Wang, "Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy," *Computers & Industrial Engineering*, vol. 113, pp. 576–588, 2017. [Online]. Available: https://doi.org/10.1016/j.cie.2017.09.050

[27] Z. W. X. H. Haiquan Sun, Wei Xia, "Agile earth observation satellite scheduling algorithm for emergency tasks based on multiple strategies," *Journal of Networks*, 2021. [Online]. Available: https://doi.org/10.1007/s11518-021-5506-4

[28] X. Liu, G. Laporte, Y. Chen, and R. He, "An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time," *Computers & Operations Research*, vol. 86, pp. 41–53, 2017. [Online]. Available: https://doi.org/10.1016/j.cor.2017.04.006

[29] R. Xu, H. Chen, X. Liang, and H. Wang, "Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization," *Expert Systems with Applications*, vol. 51, pp. 195–206, 2016. [Online]. Available: https://doi.org/10.1016/j.eswa.2015.12.039

[30] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite," *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015. [Online]. Available: https://doi.org/10.1016/j.ejor.2015.03.011.

[31] T. Mao, Z. Xu, R. Hou, and M. Peng, "Efficient satellite scheduling based on improved vector evaluated genetic algorithm," *Journal of Networks*, vol. 7, no. 3, p. 517, 2012. [Online]. Available: https://www.researchgate.net/profile/Zhengquan-Xu/publication/268352106_Efficient_Satellite_Scheduling_Based_on_Improved_Vector_Evaluated_Genetic_Algorithm/links/54d0c08e0cf298d65668859d/Efficient-Satellite-Scheduling-Based-on-Improved-Vector-Evaluated-Genetic-Algorithm.pdf

[32] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, "Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 26, 2016, pp. 345–352. [Online]. Available: https://doi.org/10.1609/icaps.v26i1.13784

[33] Y. T. Pei Wang, Peng Gao, “A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation,” 2011.

[34] J. Qi, J. Guo, M. Wang, and C. Wu, “A cooperative autonomous scheduling approach for multiple earth observation satellites with intensive missions,” *IEEE Access*, vol. 9, pp. 61 646–61 661, 2021.

[35] J. Zhang and L. Xing, “An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem,” *Computers & Operations Research*, vol. 139, p. 105626, 2022.

[36] Z. Wen, L. Li, J. Song, S. Zhang, and H. Hu, “Scheduling single-satellite observation and transmission tasks by using hybrid actor-critic reinforcement learning,” *Advances in Space Research*, vol. 71, no. 9, pp. 3883–3896, 2023.