J SYST SCI SYST ENG
Vol. 33, No. 2, April 2024, pp. 232–258
DOI: https://doi.org/10.1007/s11518-024-5591-2

ISSN: 1004-3756 (paper), 1861-9576 (online)
CN 11-2983/N

An Adaptive Local Grid Nesting-based Genetic Algorithm for Multi-earth Observation Satellites' Area Target Observation

Ligang Xing, Wei Xia, Xiaoxuan Hu, Waiming Zhu, Yi Wu

[a]School of Management, Hefei University of Technology, Hefei 230009, China
[b]Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China
[c]Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei 230009, China
xingligang@mail.hfut.edu.cn, xiawei@hfut.edu.cn (✉), xiaoxuanhu@hfut.edu.cn, zhuwaiming@mail.hfut.edu.cn, wuyi@mail.hfut.edu.cn

Abstract. The Scheduling of the Multi-EOSs Area Target Observation (SMEATO) is an EOS resource scheduling problem highly coupled with computational geometry. The advances in EOS technology and the expansion of wide-area remote sensing applications have increased the practical significance of SMEATO. In this paper, an adaptive local grid nesting-based genetic algorithm (ALGN-GA) is proposed for developing SMEATO solutions. First, a local grid nesting (LGN) strategy is designed to discretize the target area into parts, so as to avoid the explosive growth of calculations. A genetic algorithm (GA) framework is then used to share reserve information for the population during iterative evolution, which can generate high-quality solutions with low computational costs. On this basis, an adaptive technique is introduced to determine whether a local region requires nesting and whether the grid scale is sufficient. The effectiveness of the proposed model is assessed experimentally with nine randomly generated tests at different scales. The results show that the ALGN-GA offers advantages over several conventional algorithms in 88.9% of instances, especially in large-scale instances. These fully demonstrate the high efficiency and stability of the ALGN-GA.

Keywords: Multi-EOSs scheduling, area target observation, adaptive genetic algorithm, local grid nesting

1. Introduction

Earth observation satellites (EOSs) are launched for the primary purpose of imaging surface from orbit (Sun et al. 2021). In the practical applications, multiple EOSs are usually arranged to collaboratively image the area target in the scenarios such as urban planning and construction (El Garouani et al. 2017), wide-area ocean searching (Wang et al. 2020), assessment and management of agriculture (Karthikeyan et al. 2020). For example, in March 2020, a large forest fire spread rapidly in Liangshan, Sichuan Province, China. Several EOSs, including Fengyun-3, Fengyun-4, and Gaofen, were quickly deployed to collaboratively image the area affected by the forest fire in a short time.

The area target shaped like a polygon (abbreviated as "polygon") is too large to be imaged completely by an EOS in a single shot. In addition, observation requirements for the polygon involve time limits, making it difficult for a single satellite to complete a time-bounded observation requirement by taking multiple shots. As such, multiple EOSs are usually arranged to collaboratively image the polygon within a specified time (Chen et al.

2020).

The scheduling of the multi-EOSs area target observation (SMEATO) is an optimization problem. It is stated as follows: Given several heterogeneous EOSs and a large polygon waiting to be imaged, develop a workable schedule for each EOS to allow for imaging the polygon as completely as possible within a specified short time. The solutions to the SMEATO are infinite due to the variables that represent the roll-pitch angles take values from some continuous intervals. In addition, because the generated strips are strongly associated with the EOSs' imaging directions, the SMEATO is coupled to computational geometry. As such, the SMEATO is a continuous space combinatorial optimization problem coupled with computational geometry. It is nearly impossible to develop exact algorithms for finding the global optimal solutions of such a problem. In fact, the traditional EOSs scheduling problem is known to be NP-complete (Gabrel et al. 1997).

A well-known strategy for reducing the couple with computational geometry is to build a grid to discretize the area (Yuan et al. 2011). A grid-based split algorithm is developed by Zhu et al. (2019a) to first build a grid to discretize the polygon into many cells, and then generate several candidate strips according to an imaging opportunity based on the grid (Zhu et al. 2019b). However, it can be difficult to determine the most suitable gridding granularity directly. For instance, when the gridding granularity is too large, it is difficult to formulate a reasonable scheme as the generated candidate strips are too few. On the other hand, calculating a solution in an acceptable time is challenging if the polygon is divided into too many small cells. The same strip in two grids of different gridding granularity, as shown in Figure 1.

In this paper, we first consider whether it is appropriate to build a grid with the same gridding granularity to discretize the entire polygon. For example, when a part of the polygon cannot be covered by any of the given EOSs, additional grids for this part are unnecessary. An effective strategy is to divide the entire polygon into several separate local regions, so that the gridding granularity of each local region can be determined respectively. Accordingly, we propose a novel method for solving the problem. In this method, feasible solutions are formed with initial strips gained from the primary uniform rough grid, which are then improved by adding newly generated alternative strips during each iteration.

The primary contributions of this paper are as follows. 1) We propose a local grid nesting (LGN) strategy to transform the entire polygon into separate decision-making and processing components. This significantly improves the flexibility of grid division for polygons and reduces the waste of computational time caused by using a uniform small size of cells in the grid. 2) We adopt a genetic algorithm (GA) (Arbor and Holland 1975) as a framework for implementing the LGN and design an adaptive GA mutation operation for LGN (ALGN-GA). The GA's population processing and multiple iteration abilities can not only simplify the work of grid partitioning and strip construction (required for the initial solution), but also save the information from local grids and strips when they are first constructed. The ALGN-

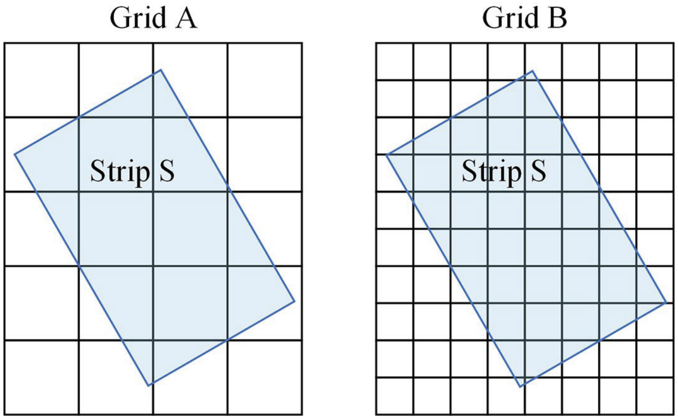


Figure 1 The Same Strip in Different Grids

GA can determine the local regions that should be retained in the current grid and those that require deep nesting. The mutation probabilities are determined based on the coverage contribution of selected strips in the parent solution and the potential increase in computational time resulting from further nesting of a given region.

The remainder of this paper is organized as follows. Section 2 provides a literature review of related works. Section 3 presents the description and modeling of an SMEATO. Section 4 introduces the LGN strategy. Section 5 describes the proposed ALGN-GA process. Section 6 discusses the validation experiments and corresponding results. Finally, Section 7 concludes the paper.

2. Literature Review

The following review of related studies is provided from the perspective of two SMEATO sub-problems: area target decomposition and scheduling optimization.

2.1 Area Target Decomposition Methods

How to divide the area is one of the most challenging problems in EOS area target observation. Hu et al. (2019) describe the observation problem using several EOSs with similar orbits, and divide the area target into several closely arranged parallel strips. Obviously, this approach of dividing the area depends on EOSs' conditions and orbital parameters, and is not suitable for multiple-EOS with different configurations. There are some EOS systems that use a pre-set positioning reference system, including the worldwide reference system (WRS) used by the U.S. in planning Earth observation missions for the Landsat series EOSs and the grid reference system (GRS) used by the French SPOT EOSs. However, these predefined reference systems generally only consider the areas in which the EOS is looking directly at the ground without taking the EOSs' rolling capabilities into account. As such, the segmentation of area targets using a reference system often limits EOS imaging capabilities and does not take full advantage of remote sensing resources (Zhang et al. 2012).

A review of previous studies suggested that dividing the area based on the grid is a common strategy for multi-EOSs problems. The most significant advantage of this strategy is that it can decouple the EOS trajectory from the location of the target area with low computational cost. Li et al. (2006) decompose area targets based on grids with the same intervals of latitude and longitude. Zhu et al. (2019b) design a grid-based split technique that offers higher flexibility of constructing strips by referring to the grid locations. The granularity of the grid will obviously affect the coverage accuracy when using the grid-based dispersion method. However, it is determined empirically in most practical applications (Zhang et al. 2012). Notably, Hu et al. (2021) propose a nested father-child structure, in which the smaller-grained grid is not acquired by direct division but by subdividing the larger-grained grid. However, they only test the nested grid once with only a single pair of father-child grids, but do not conduct in-depth verification of nesting effects. In this paper, the grid-based split and father-child structures are used to conduct a further investigation of gridding granularity for area targets.

2.2 Scheduling Optimization Algorithms

Several algorithms have been applied to the EOS scheduling. For example, Lemaître et al. (2002) compare the effects of greedy, dynamic programming, constrained programming, and local search algorithms to solve regional problems, providing a discussion of different usage scenarios for each algorithm. Gabrel and Vanderpooten (2002) use the enumeration to define a feasible and satisfactory shot sequence for EOS. Perea et al. (2015) model the multi-EOSs area targets problem as a set of coverage tasks and solve it with the greedy constructive algorithm. In addition, some other prominent local search algorithms, such as simulated annealing algorithm, tabu search algorithm, are also employed in EOS scheduling (Peng et al. 2011, Sarkheyli et al. 2013, Wang et al. 2015). However, these algorithms either have a high computational cost or can easily lead to suboptimal results.

Bio-inspired algorithms, which usually involve some form of intelligence, are ideal alternatives to solve the combinatorial optimization problems. Dorigo and Gambardella (1997) introduced the Ant Colony Optimization (ACO) algorithm as a novel approach based on the foraging behavior of real ant colonies, which has become a popular and effective method for solving a wide range of combinatorial optimization problems, including the Traveling Salesman Problem. Zhang et al. (2014) propose a novel approach for multi-satellite control resource scheduling using ant colony optimization. They address the issue of allocating resources, such as power, bandwidth, and memory, to a group of satellites while satisfying various constraints, such as communication link quality, mission objectives, and equipment limitations. The problem is formulated as a multi-objective optimization task, and ant colony optimization, a nature-inspired optimization technique, is used to obtain the optimal solution. Karaboga and Basturk (2007) introduce the Artificial Bee Colony (ABC) algorithm, a metaheuristic optimization approach inspired by the foraging behavior of honeybees, which has demonstrated high effectiveness across various optimization problems. Luo (2020) presents a hybrid binary artificial bee colony algorithm for solving the satellite photograph scheduling problem. The algorithm combines the strengths of binary encoding and artificial bee colony algorithms to optimize the scheduling of satellite photographs while considering various constraints, such as the availability of satellites and the need to cover specific areas at specific times. These studies demonstrate the potential of bio-inspired algorithm for solving complex optimization problems.

The genetic algorithm, a well-known bio-inspired algorithm, was first developed by Arbor and Holland (1975). As a global optimization probability search technique, the GA has the dual advantages of parallel search based on exchanges within the population and new features generated by natural mutation. As such, it has been widely adopted to solve various combinatorial optimization problems, including EOS scheduling. Song et al. (2019) use GA to optimize the generated task sequence in the satellite image downlink scheduling problem to achieve higher profit. Zhou et al. (2015) apply GA to the multiple geosynchronous EOS inspection problem. And Zhu et al. (2019a) use GA to investigate an integrated approach to Earth observation satellite scheduling.

In addition to effective crossover and mutation strategies, the probability of these events is also crucial in determining the efficiency of GA. In traditional GA, mutation probability is an empirical value that remains fixed. However, if the mutation probability is too small, the population will converge to a local optimum due to a lack of diversity. And it will be difficult to preserve the good traits of better solutions when the mutation probability is too large. Srinivas and Patnaik (1994) propose an adaptive GA (AGA), in which the probability of crossover and mutation will increase when the fitness of individuals is low and decrease when the fitness of individuals is high. Some researchers have used AGAs to solve EOS area target observation problems. Li et al. (2019) introduce an activation sigmoid function into the calculation of crossover and mutation probability to optimize the rolling angle for a lunar EOS in an area imaging task. Computational experiments demonstrate that AGAs exhibit obvious advantages over traditional GA in searching for optimal results. This paper also utilizes adaptive parameter adjustments to determine mutation probabilities.

3. Problem Formulation

3.1 Problem Description

Figure 2 illustrates the scenario investigated by the SMEATO problem, which involves an imaging requirement for an earth area of polygonal shape and multiple EOSs, each equipped with a visible light camera capable of scanning and imaging the Earth's surface. However, the size of the polygonal area is larger than the imaging scope of an EOS, and thus, it cannot be fully imaged in a single pass. The camera onboard the EOS has a rolling angle range, which defines the maximum rolling angle when it swings to its extreme position. By swinging to the maximum rolling angle in both directions, the camera can obtain a maximum image field for each imaging opportunity, represented by the areas denoted by the dotted lines in Figure 2. This image field is called the field of view and represents the largest range that can be scanned within a given time window. However, due to the limited viewing angle of the camera, each polygon request must be divided into strip areas (abbreviated as "strips"), represented by rectangles on the plane of the field of view, which correspond to the imaging range of an EOS scan.

The curve formed by the intersection of a straight line, drawn from the orbit of the EOS to the centre of the earth, and the Earth's surface is called the ground track. The strips may traverse the ground track through its centre or not intersect it, but must always remain parallel to the corresponding ground track. The time period during which an EOS flies over or near the polygon and is able to effectively image the target is called visible time window. It is important to note, however, that there is a maximum imaging time span for each EOS, and it cannot take continuous photos for an extended period during a single imaging action.

Since the camera on the EOS is capable of taking pictures at various rolling angles, each EOS has the freedom to determine which part of a polygon can be imaged during an imaging opportunity. Therefore, the location of the strip imaged by an EOS during an imaging opportunity is variable. Furthermore, the length of the strip is not fixed but depends on the coverage start and end times. Consequently, multiple candidate strips exhibiting different coverages could be acquired for each opportunity. A set of strips (with different start times, end times, and rolling angles) acquired during the same imaging opportunity is called a brother strip set. The strips in a brother strip set are mutually exclusive, meaning that once a strip is selected, the corresponding opportunity is occupied along with the determined angle and time slot.

The SMEATO problem can be summarized as satisfying the imaging request as completely as possible with limited EOS resources. Specifically, we need to compose an imaging plan that determines the rolling angle of each EOS over the polygon for each visible time window, as well as the imaging start and end times. Observation strips determined by the imaging plan form a final coverage scheme. The objective of the problem is to make full use of inadequate EOS resources to cover the maximum area in the polygon. Thus, the maximum percentage of coverage for the polygon is used as the optimization objective.

To compose such scheme, some realistic constraints should be considered including: 1) The rolling angle of the camera onboard each EOS is less than its corresponding maximum rolling angle. 2) Each EOS can only image a single strip in one opportunity. 3) Each strip must be imaged during its corresponding visible time window. 4) The continuous imaging duration of each EOS is less than the corresponding maximum imaging time span during a single imaging action.

In practical applications, the execution of imaging tasks by EOSs is subject to several restrictions such as energy, storage, and data download limitations. In cases where energy or storage surpasses these limits, targets cannot be observed, and mission completion is impossible when the satellite's image data cannot be downloaded promptly. Consequently, this paper adopts the following realistic assumptions: 1) each EOS carries only one camera; 2) storage and energy are sufficient; 3) ground stations have sufficient antenna resources to receive EOSs' image data at a given time; 4)

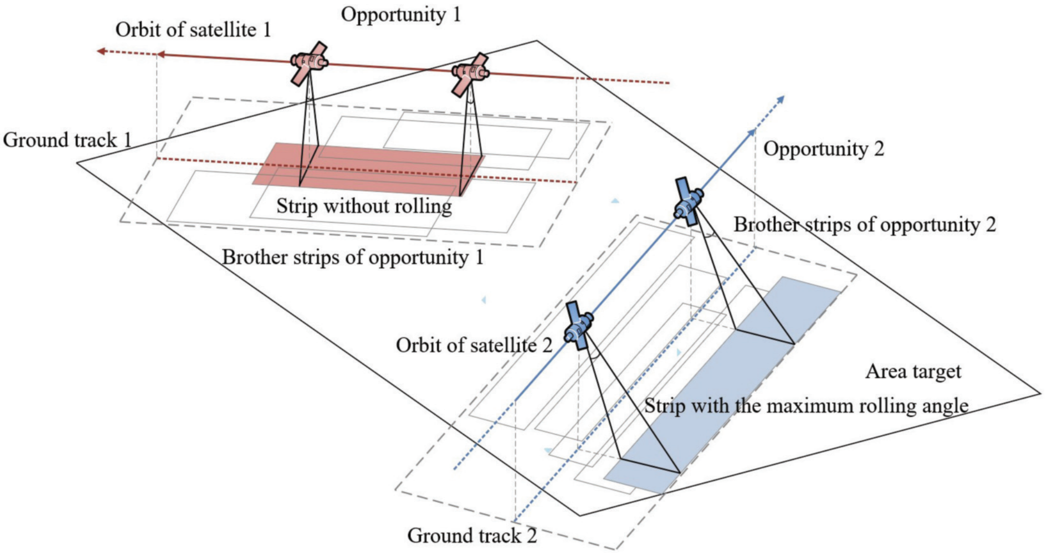


Figure 2 Area Target Coverage from Multiple EOSs

although EOS ground tracks are curved, they can be approximated as tangential paths during short imaging times. These assumptions are primarily based on the characteristics of the Chinese Gaofen-series satellites and have been extensively utilized in related literature.

3.2 Mathematical Model

The problem has a complicated objective and constraints. We develop a constraint optimization formulation to model the problem. The notations used in the model and their definitions can be found in Table 1.

The problem aims to optimize the use of limited EOS resources by maximizing the coverage of a given polygon. The actual coverage of the polygon is the union of all selected strips for all imaging opportunities. Let r_{ijk} be the area for s_{ijk} and f be the actual coverage of the polygon. A decision variable x_{ijk} is used to denote whether the strip s_{ijk} is selected. If the strip $s_{ijk} \in S_{ij}$ is selected, $x_{ijk} = 1$; Otherwise, $x_{ijk} = 0$. The optimization objective is defined as the maximum percentage of polygon coverage. Then, we obtain the following equation:

$$f = \max \cup_{i=1}^{|F|} \cup_{j=1}^{|C_i|} \cup_{k=1}^{|S_{ij}|} r_{ijk} \times x_{ijk} \tag{1}$$

The rolling angle of the camera onboard each EOS is less than its corresponding maximum rolling angle, as follows.

$$|\omega_{ijk}| \leq W_i, \quad \forall i \in I, \forall j \in J, \forall k \in K \tag{2}$$

The restriction, which requires each EOS to image no more than one strip for each opportunity, is expressed as follows.

$$\sum_{k=1}^{|S_{ij}|} x_{ijk} \leq 1, \quad \forall i \in I, \forall j \in J, \forall k \in K \tag{3}$$

Each strip must be imaged during its corresponding visible time window, which is given as follows.

$$\begin{cases} a_{ijk} \leq b_{ijk} \\ e_{ijk} \leq d_{ijk} \end{cases} \quad \forall i \in I, \forall j \in J, \forall k \in K \tag{4}$$

The continuous imaging duration of each EOS is less than the corresponding maximum imaging time span during a single imaging action.

$$0 \leq e_{ijk} - b_{ijk} \leq T_i, \quad \forall i \in I, \forall j \in J, \forall k \in K \tag{5}$$

Table 1 The Table of Notations that Used in the Formulation

Notation	Definition
i	EOS index, $i \in I = \{1, 2, \cdots, \|F\|\}$, where $\|F\|$ is the total number of EOSs
F_i	the $i-th$ EOS, $F_i \in F = \{F_1, F_2, \cdots, F_{\|F\|}\}$
j	imaging opportunity index, $j \in J = \{1, 2, \cdots, \|C_i\|\}$, where $\|C_i\|$ is the total number of the imaging opportunities for F_i
c_{ij}	the $j-th$ opportunity for F_i, $c_{ij} \in C_i = \{c_{i1}, c_{i2}, \cdots, c_{i3}\}$
k	strip index, $k \in K = \{1, 2, \cdots, \|S_{ij}\|\}$, where $\|S_{ij}\|$ is the total number of strips for c_{ij}
s_{ijk}	the $k-th$ strip for c_{ij}, $s_{ijk} \in S_{ij} = \{s_{ij1}, s_{ij2}, \cdots, s_{ij\|S_{ij}\|}\}$
W_i	the maximum rolling angle for the camera onboard F_i
V_i	the viewing angle for F_i
T_i	the maximum imaging time span for F_i
o_{ij}	the ground track segment for c_{ij}
v_{ijk}	the visible time window corresponding to c_{ij}
a_{ijk}	the start time for v_{ijk}
d_{ijk}	the end time for v_{ijk}
b_{ijk}	the start time for imaging s_{ijk}
e_{ijk}	the end time for imaging s_{ijk}
ω_{ijk}	the rolling angle corresponding to s_{ijk}
r_{ijk}	the area for s_{ijk}
l_{ijk}	the left boundary line of s_{ijk}
x_{ijk}	a decision variable. If the strip s_{ijk} is selected, $x_{ijk} = 1$; Otherwise, $x_{ijk} = 0$

Equations (1)-(5) constitute the constraint optimization formulation for the primal SMEATO problem, which aims to compose an imaging plan for limited EOS resources that covers the maximum area in a polygon. The optimization objective is to achieve the maximum percentage of coverage for the polygon. The model considers several realistic constraints, such as the maximum rolling angle of the camera onboard each EOS, the ability of each EOS to image a single strip in one opportunity, the requirement that each strip must be imaged during its corresponding visible time window, and the limitation on the continuous imaging duration of each EOS.

As described, each imaging strip corresponds to a variable, and all variables take values from continuous intervals. Different start-stop imaging times and roll-pitch angles result in considerable variation in the lengths and locations of the strips for each imaging opportunity. The problem is obviously a nonlinear combinatorial optimization problem in continuous space. Moreover, the generated strips are strongly associated with the imaging directions of the EOSs. Because the polygon has multiple imaging opportunities in different directions due to the heterogeneity of the EOSs, the SMEATO is coupled with computational geometry. It is nearly impossible to develop efficient exact algorithms to find the global optimal solutions of the primal SMEATO problem. Therefore, the SMEATO described in this paper is much more challenging than the traditional EOS problem, which only considers the optimization of strip selection.

4. Grid-based Method of Discretization

This section provides detailed descriptions of the grid-based discretization method used to solve the SMEATO problem. As a continuous space combinatorial optimization problem coupled with computational geometry, the SMEATO theoretically has an infinite number of values for variables. However, executable solutions require discrete variable values and a solution space with a limited number of combinations. In this paper, the widely used grid discretization technique is adopted to discretize the polygon into a grid with many square units, each of which forms a cell. The corresponding finite strips construction process is described as follows.

4.1 Grid-based Split

The strip construction technique used in this paper is adapted from a method developed by (Zhu et al. 2019a). As shown in Figure 3, the grid, denoted as G, comprises many square cells. Three anchor cells called the left cell $t \in G$, the top cell $g \in G$, and the bottom cell $u \in G$ are established and used to determine the left, upper, and lower boundaries of each feasible strip. Anchor cells can be reused, either for a single strip or for different strips. The Figure 3 shows a strip from the top left to the bottom right on the left, and another one from the top right to the bottom left on the right.

Taking the left strip as an example, when the opportunity c_{ij} and the corresponding ground track segment o_{ij} are known, a cell can be determined by forming a line which is parallel to o_{ij} and pass through the lower left vertex of the cell. Furthermore, the determined cell must be in the maximum image field for opportunity c_{ij} so that the maximum rolling angle constraint of the camera onboard the EOS is satisfied. The formed line is denoted as l_1 and serves as the left boundary of the new strip. And then, the rolling angle can be uniquely determined after the left boundary l_1 is identified. Let d_t represents the distance from l_1 to o_{ij} and h_i be the altitude of the EOS F_i. The width w_t of the corresponding strip can be calculated by equation (6).

$$w_t = \begin{cases} d_t - h_t \times \tan(\arctan(d_t/h_i) - V_i), \\ \qquad\qquad \text{if} \arctan(d_t/h_i) > V_i \\ d_t + h_t \times \tan(V_i - \arctan(d_t/h_i)), \\ \qquad\qquad \text{if} \arctan(d_t/h_i) \leq V_i \end{cases} \tag{6}$$

After the line l_1 and the strip width w_t are calculated, the position of the line l_2, which is the right boundary of the strip, can also be determined. Finally, a pair of cells, g and u, can be identified between the parallel lines l_1 and l_2. Both the line l_3 and l_4 are perpendicular to o_{ij} and are drawn through the upper left vertex of cell g and the lower right vertex of cell u, forming the upper and lower boundaries of the strip, respectively. The number of anchor cell triplets in the grid that conform to these relationships dictate the maximum number of alternative strips. The pseudocode of the improved grid-based split method can be found in Algorithm 1.

Furthermore, while the generated strips are allowed to overlap, each newly generated strip should be compared with existing strips in the same opportunity to eliminate redundancy in

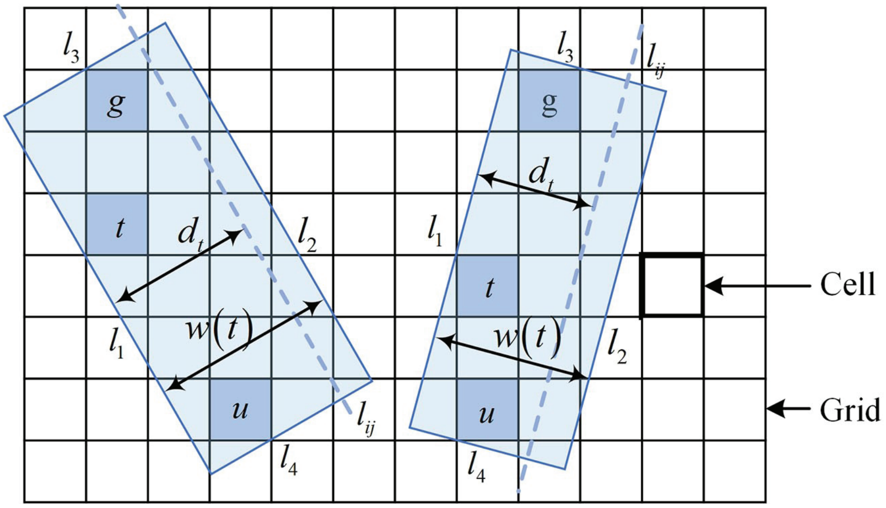


Figure 3 The Grid, Cells, and Grid-based Strip Generation

Algorithm 1 Grid-based splitting

```
Input: covering opportunity $c_{ij}$ and grid $G$
Output: candidate strips set $S_{ij}$
for $m = 1 \rightarrow |G|$ do
    /*$t_m$ is the $m - th$ cells in $G$ */
    Calculate $l_1$ and $w(t_m)$ according to $t_m$
    Calculate $l_2$
    /*$G'$ is the part of $G$ between the parallel lines $l_1$ and $l_2$ */
    for $n = 1 \rightarrow |G'|$ do
        for $o = 1 \rightarrow |G'|$ do
            /*$g_n$ and $u_o$ are the $n - th$ and $o_th$ cells in $G'$, respectively */
            if $< t_m, g_n, u_o >$ satisfy conditions then
                generate $\alpha$ according to $< t_m, g_n, u_o >$
                $S_{ij} \leftarrow \alpha$
            end if
        end for
    end for
end for
return candidate strips set $S_{ij}$
```

strip sets. If the covering area of one strip is a subset of another, the included strip will be deleted. Polygon Boolean operation proposed by Vatti (1992) is adopted to calculate the union of selected strips and determine the coverage of solutions in this paper.

4.2 Analysis of Difficulties

As a form of approximation, the granularity of grid discretization affects both computational complexity and results. Finer precision produces a more optimal solution but increases the solution space, while lower precision reduces computational complexity and running time but sacrifices solution optimality. The influence of the number of cells in the grid on solution quality and computational cost are discussed below.

1) Theoretically, with maximum coverage as the optimization goal, increasing the number of cells will inevitably improve results, as the combinatorial optimization from larger spaces will never be worse than that of smaller spaces.

2) Constructing one strip requires a three-layer traversal to search the grid for anchor cell

triplet. The first layer traverses all cells in the grid to find the left cell t, and the next two layers traverse the cells between the parallel lines l_1 and l_2 to find the top cell g and the bottom cell u. Let q be the average number of strips generated from an opportunity, eliminating redundant strips then requires comparing each newly constructed strip with existing strips of the same opportunity, and a total of $\frac{q(q-1)}{2}$ comparisons are required to construct a strip set.

3) For a given opportunity c_{ij} increasing the number of cells generates a greater number of triplets of anchor cells that meet these conditions, thereby enlarging the set of candidate strips. To identify a strip from each brother strip set, we can easily figure out that the total number of all feasible combinations is $|S_{ij}|$. Therefore, as the number of candidate strips for c_{ij} increases, the total number of all feasible combinations $\prod_{i=1,j=1}^{i=|F|,j=|C_i|} |S_{ij}|$ correspondingly increases, and thereby results in an expansion of the solution space.

In summary, the number of cells and strips are key factors in determining the required number of calculations and the quality of a solution, and the number of strips is closely related to the number of cells. Therefore, this paper proposes a local nesting strategy to resolve the dilemma between solution results and computational complexity, by indirectly controlling the number of cells.

4.3 LGN Based on a Father-child Grid

In our proposed LGN strategy, grids with different sizes of cells coexist and the optimization process for strip combination start from a grid with small number of large cells. Sections of target area are then gradually divided into additional grids with smaller cells, thereby providing more candidate strips and expanding the solution space in a controlled manner.

4.3.1 Father-child Grids

The father-child grid is proposed by Hu et al. (2021) to speed up solutions for sub-problems in column generation-based processes. In this paper, we improve the technique with a novel LGN strategy. In father-child grids, each father cell from the father grid can completely cover its child cells in the child grid. If a nesting operation shortens the side length of a cell to $1/2$ of its original size, the father cell will be divided into 4 child cells. If the length is shortened to $1/3$, one father cell will produce 9 child cells. Child cells can also be divided in the same way. When there are more than two nesting layers, every two adjacent grid layers form a pair of father-child grids, and the new cells will be completely covered by the original father cell. Let P be the grid layer, which increases with nesting operations, $P = 1, 2, 3$.

Father-child grids have two characteristics. First, the father cells completely include child cells, which ensures different layers of grids and strips can coexist. Second, candidate strips constructed in father grids are also feasible in child grids. This implies that in a given area, the set of strips constructed by child cells completely contains the set of the strips constructed by father cells. In other words, nesting the grid means expansion of the original set. According to these characteristics, this paper applies the nesting of father-child grids to the local grid area, and proposes the LGN strategy.

4.3.2 Nesting of Local Regions

In this paper, the maximum image field (defined in Section 3.1 and indicated by the dotted lines in Figure 2 and Figure 4) for opportunity c is considered to be a local region, and denoted as LR_c. These types of local regions, divided based on opportunities, are allowed to overlap and do not affect each other. In LGN, father-child grids are applied to local regions. Instead of dividing the entire polygon area into a grid with the same granularity, some parts of the grid are divided deeper with smaller granularity while other parts are kept at a higher layer with a larger granularity. A triplet of anchor cells can be found in the same layer and used to form a strip. The strips on different layers coexist in a grid are shown in Figure 4, where the brighter cells indicate anchor cells for each strip.

LGN can be conducted in two steps: nesting the local grid of LR_c and constructing new strips for c in the nested grid. The details of these two steps are as follows.

Step 1. Nesting the local grid of LR_c

Initially, the entire area is evenly divided into a grid with the same granularity. A set of brother strips can be constructed for each opportunity in the first layer of the grid. Then a feasible and smaller solution space can be developed accordingly. Constructing new strips for opportunity c requires dividing each of the cells in the local region LR_c to acquire corresponding child cells. The selection of an opportunity for nesting of the local region can be incorporated into the algorithm with a GA framework and, more specifically, into the mutation operation. The details of this selection method are discussed in Section 5.2.

Step 2. Constructing new strips for c in the nested grid

After acquiring child cells for LR_c, new strips are generated for c using the child cells, as discussed in Section 4.1. The strips in layer P are denoted as S_k^P and the set of brother strips for opportunity c can be expressed as $S^P = \{s_1^P, s_2^P, \cdots, s_m^P\}$ where m denote the total number of strips for c . Local grid nesting of one opportunity does not affect the strips, including the anchor cells, of other opportunities. For example, if LR_{c_1} and LR_{c_2}, which correspond to different imaging opportunities (c_1 and c_2), are overlapping, the part of the overlap can also be divided when nesting LR_{c_1}. But LR_{c_2} remains unchanged and generates no new strips for c_2 in this process, unless the nesting of LR_{c_2} is performed.

5. ALGN-GA

The proposed LGN strategy is combined with an AGA to form the ALGN-GA. This section describes chromosome encoding, the generation of initial solution for the SMEATO problem, a novel adaptive mutation operator, and the crossover operator for the ALGN-GA. Figure 5 illustrates the flowchart of the ALGN-GA.

5.1 Encoding and Initialization

The first step in the ALGN-GA is to represent the solution as a chromosome. Obviously, if $P_1 \neq P_2$, then $|S_{ij}^{P_1}| \neq |S_{ij}^{P_2}|$. In addition, $|s_{ijk}^{P_1}|$ and $|s_{ijk}^{P_2}|$ represent two different strips, even for the same k . Therefore, it is necessary to store the strip layer and strip ID together at the corresponding chromosome position to uniquely identify the selected strip. As shown in Figure 6, a double-layer chromosome is proposed for the development of a feasible coverage scheme. The length of a chromosome

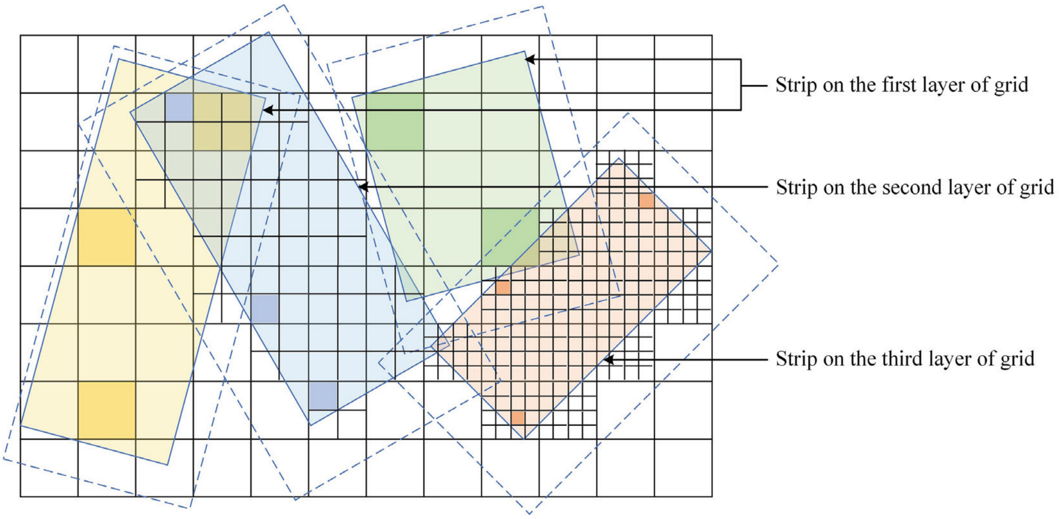

Strip on the first layer of grid
Strip on the second layer of grid
Strip on the third layer of grid

Figure 4 Local Grid Nesting and Multi-layer Strips

represents the number of imaging opportunities. Each opportunity equips a constant ID corresponding to the order in which they are provided. The strip selected for the opportunity are represented by a pair of arrays at corresponding locations.

The numbers in the first row represent the strip layer and the numbers in the second row represent the strip ID. Deeper grid layers are indicated by darker colors in Figure 6. This chromosome represents a set of strips $\{s^1_{ij6}, s^1_{ij4}, s^3_{ij52}, s^2_{ij21}, s^2_{ij2}, s^1_{ij5}, s^3_{ij32}, s^1_{ij11}, s^4_{ij40}, s^1_{ij7}\}$, which is a coverage scheme. The fitness is the coverage of the solution which is calculated by the ratio of the union area for all strips to the area of the target area. This encoding strategy can give a fixed chromosome length and avoid the redundant information. In the ALGN-GA, the construction of initial solutions is performed randomly. First, candidate strips from all opportunities are constructed in the first layer father grid, and a random strip is then selected from each opportunity to form an initial solution.

5.2 Adaptive Mutation Operator

5.2.1 Mutation Operator

GA mutation operators can be used to generate new characteristics that have not appeared in the existing population to increase diversity. In this paper, the role of mutation is not only to improve initial solutions, but also to conduct LGN. In each iteration, a random parent individual F is first selected and represented as a chromosome. Then, an opportunity $c_{ij} \in F$ is selected, along with its accompanying strip $s^P_{ijk} \in S^P_{ij}$, and $c_{ij} \in F$ can either result in a mutation or not. If a mutation occurs, each cell in $LR_{c_{ij}}$ is divided into child cells, and new strips are constructed. The set of brother strips S^{P+1}_{ij} can then be acquired, and a new strip for c_{ij} will be randomly selected from the set S^{P+1}_{ij} to replace the original strip $s^P_{ijk} \in S^P_{ij}$. If no mutation occurs, a new strip will be selected from the original brother strip set S^P_{ij}. Figure 7 shows an example of a mutation operator. Mutation of the parent chromosome occurs at the seventh opportunity. In this process, the strip selected by the parent chromosome is denoted No. 2 in the set S^1, while the strip selected by the offspring chromosome is strip No. 27 in the set S^2 after mutation.

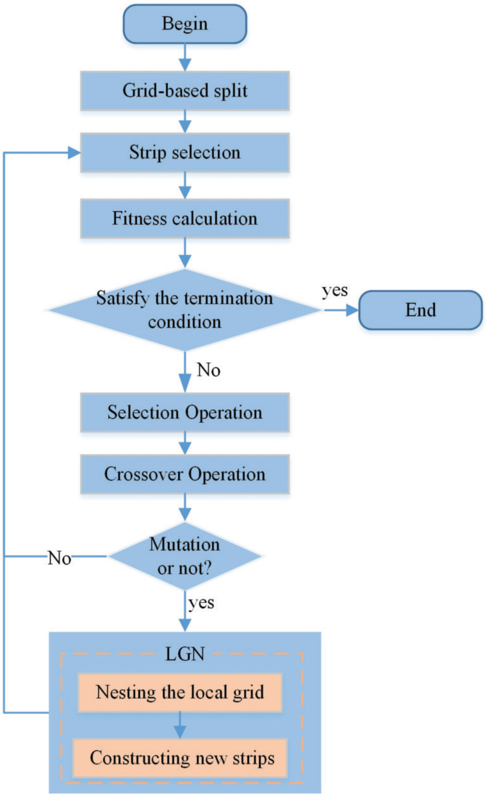


Figure 5 Flowchart of the ALGN-GA

It is worth noting that in the ALGN-GA, the preservation and sharing of population genes results in the nesting of layers for a single opportunity, which occurs no more than once during the iterative process. The resulting strip information can be shared in every population. In other words, once LGN is initially conducted for a c_{ij} from layer P to $P+1$ during the mutation of a solution, S_{ij}^{P+1} will be preserved. When the same mutation occurs for another solution, S_{ij}^{P+1} can be acquired from the existing information and there will be no need to reconstruct a new $P+1$ layer strip for c_{ij} This is one of the primary reasons for combining LGN with the GA.

5.2.2 Adaptive Probability Function

In the first step discussed in Section 4.3, the nesting of grids is performed in local regions, and it is evident that individual areas can be divided at different scales and speeds. Otherwise, local nesting is meaningless. As such, determining the optimal degree of nesting for each local region is a crucial decision. In this paper, the performance of a selected strip s_{ijk}^P is used to determine whether the opportunity c_{ij} required further nesting of $LR_{c_{ij}}$, as part of a brother strip set update from S_{ij}^P to S_{ij}^{P+1}. As such, we develop an adaptive probability mutation function involving the following definitions:

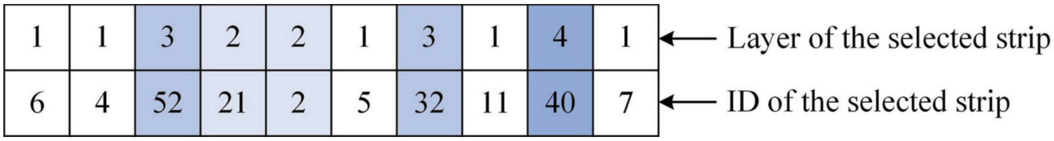
Definition 1 - Unique Coverage Area (UCA): A certain part of the target area $r \in R$ covered only by a single strip s. The area of r is the unique coverage area for the strip s and is denoted UCA_s.

Definition 2 - Efficient Strip Rate (ESR): The ratio of UCA_s to the area of strip s, denoted ESR_s.

Definition 3 - Actual Covering Rate (ACR): The ratio of UCA_s to the area of the target area R, denoted ACR_s.

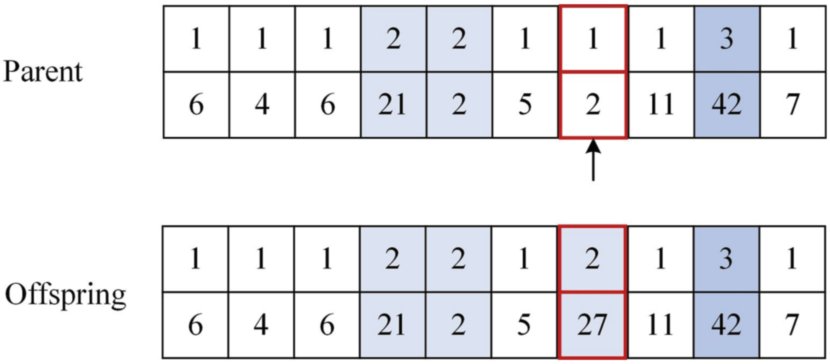
Definition 4 - Scale Expand Magnification (SEM): The ratio of the number of brother strips in layer P to the number of brother strips in layer $P-1$: $|S^P|/|S^{P-1}|$. This value for the layer P and the opportunity c is denoted SEM_c^P. Note that $SEM_c^P \equiv 1$, if $P = 1$.

For solutions consisting of multiple strips, UCA_s is the true value of a strip's contribution to the coverage of a target area for the solution. Two additional indicators, ESR_s and ACR_s, are introduced to represent the effective proportion of strip s and its contribution to the



1	1	3	2	2	1	3	1	4	1
6	4	52	21	2	5	32	11	40	7

Figure 6 An Example of the Double-layer Chromosome



Parent	1	1	1	2	2	1	1	1	3	1
	6	4	6	21	2	5	2	11	42	7
Offspring	1	1	1	2	2	1	2	1	3	1
	6	4	6	21	2	5	27	11	42	7

Figure 7 An Example of a Mutation Operation

entire solution, respectively. Higher values of these two indicators imply a higher probability of maintaining the current strip.

Furthermore, the number of strips generated from a certain opportunity depends on several parameters, including the viewing angle, the maximum rolling angle for the corresponding EOS, and the relative position between the EOS trajectory and the target area. These factors affect the number of strips through geometric relationships, both directly and indirectly. As such, we introduce SEM to demonstrate the viability of further local region nesting. The value of SEM_c^P can provide a reference for opportunity c in advancing layers from P to $P+1$. A large SEM_c^P value indicates the nested opportunity c is likely to have a significant calculation impact and tends to reduce the possibility of nesting or generating a new strip set S^{P+1}.

Taking these indicators into consideration, the adaptive mutation probability P_{CSk} can be calculated as follows:

$$P_{CSk} = \begin{cases} 0.5 \times e^{-ESR_S} + 0.2 \times e^{-ACR_S} \\ +0.3 \times \dfrac{\ln\left(SEM_c^P + e - 1\right)}{SEM_c^P + e - 1}, & P < P_{\max} \\ p_0, & P \geq P_{\max} \end{cases} \tag{7}$$

where $P_{\max}$ is the pre-established maximum layer, whose value is affected by the actual size of the problem, including the area of target, the number of EOSs and opportunities. If the current strip layer has reached $P_{\max}$, it will no longer be nested, thus p_0 is set to 0. It is important to highlight that the weights "0.5, 0.2, 0.3" were determined based on empirical experience and extensive simulation testing, which involved multiple rounds of experimentation. These weight values therefore represent our best estimates for optimizing the algorithm's performance.

For $P < P_{\max}$, $ESR_S \in (0,1]$ and $e^{-ESR_S} \in [1/e, 1]$, such that $0.5 \times e^{-ESR_S} \in [1/(2e), 1/2]$. Similarly, $0.2 \times e^{-ESR_S} \in [1/(5e), 1/5]$. In addition, from $SEM_c^P \in [1, +\infty)]$, we can calculate that $\frac{\ln(SEM_c^P + e - 1)}{SEM_c^P + e - 1} \in (0, 1/e]$. And finally,

$P_{CSk} \in [7/(10e), 7/10 + 3/(10e)) \subset (0,1)$. The corresponding adaptive mutation operation is shown in Algorithm 2.

Algorithm 2 The adaptive mutation operation

```
Input: A parent solution $F$
Output: An offspring solution $O$
Propose a new solution $O = F$
Randomly select an opportunity $c$ and the corresponding strip $s \in O$
Identify the strip set $S^P$
$O \leftarrow \{O \setminus s\}$
Generate a new set $S' \leftarrow S^P$
if $P < P_{\max}$ for strip $s$ then
    Calculate the mutation probability $P_{CSk}$
    Generate a random number $p' \in (0,1)$
    if $p' \leq p_{jk}$ then
        if this is the first nesting of opportunity $c$ from layer $P$ to $P+1$ then
            Divide the cells within $LR_c$
            Generate a new strip set $S^{P+1}$
            $S' \leftarrow S^{P+1}$
        else
            Find the strip set $S^{P+1}$ from the saved information
            $S' \leftarrow S^{P+1}$
        end if
    end if
end if
Randomly select a strip $s'$ from $S'$
$O \leftarrow \{O \setminus s'\}$
return the offspring solution $O$
```

5.3 Crossover Operator

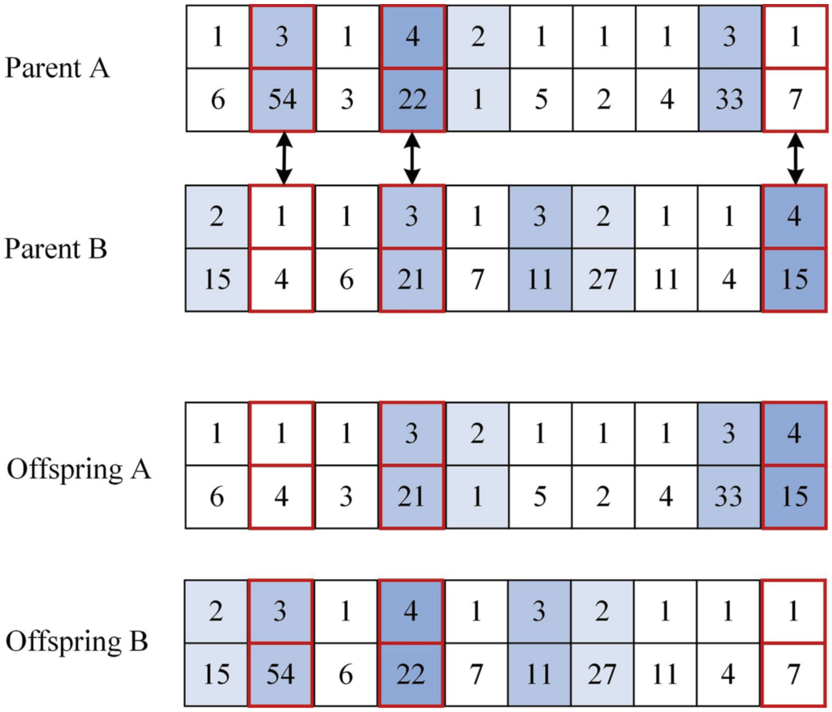
In most cases, crossover operations aim to inherit favorable traits from the population's existing individuals. For this objective, a multi-point crossover operator is adopted in this paper. In each iteration, two solutions are randomly selected as parent individuals from the current population, denoted F_A and F_B. Several random strip pairs are then identified and exchanged between the parents. The number of strips contained in F is denoted $|F|$, and that $|F_B| = |F_A|$. The number of strips selected for exchange is set to $\alpha = \lfloor 0.2 \times |F_A| \rfloor + 1$, where 0.2 was also determined based on empirical knowledge and extensive simulation testing. Figure 8 shows an example of the crossover operation that generates two new individuals, the offspring O_A and O_B, from the selected parents. The pseudocode for this crossover operation is provided in Algorithm 3.

6. Computational Experiments

This section presents a series of computational experiments and analyses of the experimental results. Section 6.1 discusses the settings of nine instances with various grid and opportunity scales. The process and results for the experiments of ALGN-GA are documented in Section 6.2. And Section 6.3 demonstrate the effectiveness of the LGN. In Section 6.4, the ALGN-GA is compared with several representative algorithms, each of which is programmed in C# and tested on a PC with 24GB of RAM and an Intel Core i7-7700 processor running a 3.60 GHz 64-bit Windows 10 operating system.

6.1 Instance Settings

Since there is no acknowledged benchmark dataset for this problem, a series of test instances are randomly generated. As seen in Table 2, nine instances are produced with different initial grid and varying numbers of EOSs and opportunities. The initial grids are ranging from 3×3 to 30×30, which cover rectangles and squares. And as the area of a target area



Parent A

1	3	1	4	2	1	1	1	3	1
6	54	3	22	1	5	2	4	33	7

Parent B

2	1	1	3	1	3	2	1	1	4
15	4	6	21	7	11	27	11	4	15

Offspring A

1	1	1	3	2	1	1	1	3	4
6	4	3	21	1	5	2	4	33	15

Offspring B

2	3	1	4	1	3	2	1	1	1
15	54	6	22	7	11	27	11	4	7

Figure 8 An Example of a Crossover Operation

Algorithm 3 The crossover operation

```
Input: Two parent solutions $F_A$ and $F_B$
Output: Two offspring solutions $O_A$ and $O_B$
Propose a new solution $O_A = F_A$
Propose a new solution $O_B = F_B$
Let $\alpha = \lfloor 0.2 \times |F_A| \rfloor + 1$
Generate $\alpha$ random numbers $\lambda_1, \lambda_2, ..., \lambda_\alpha \in \{1, 2, ..., |F_A|\}$
for each $\lambda \in \{\lambda_1, \lambda_2, ..., \lambda_\alpha\}$ do
    $O_A \leftarrow \{O_A \setminus s_{A\lambda}\}$ ($s_{A\lambda}$ represents the strip of opportunity $c_\lambda$ in $O_A$)
    $O_A \leftarrow \{O_A \setminus s_{B\lambda}\}$ ($s_{B\lambda}$ represents the strip of opportunity $c_\lambda$ in $O_B$)
    $O_B \leftarrow \{O_B \setminus s_{B\lambda}\}$
    $O_B \leftarrow \{O_B \setminus s_{A\lambda}\}$
end for
return offspring solutions $O_A$ and $O_B$
```

increases, the number of EOSs increases from 3 to 7 and the number of corresponding opportunities increases from 3 to 30.

6.2 Experiment of ALGN-GA

In the experiments of ALGN-GA, the population size for each generation is set to 10, and the maximum running iteration for each test is set to 100. Tests of each instance are repeated 20 times independently. And then, the running time for each algorithm is limited to less than 1200 seconds (20 minutes). The process is terminated and invalidated if the algorithm could not complete the operation within this time. Algorithms that did not terminate within the limit time are recorded as "-" in the tables.

Table 3 shows the best solution acquired by the ALGN-GA for each instance, in which "$P_{\max}$" indicates the maximum layer set. Our optimization goal is to maximize the resulting solution coverage rate. The highest, lowest, mean, and standard deviation for 20 repeated tests are labeled as "Max", "Min", "Mean", and "SD", respectively. Additional details for

Table 2 Random Test Information for Instances I1-I9

Instance	Initial grid	Number of EOSs	Number of opportunities
I1	3×3	3	3
I2	3×5	3	4
I3	5×5	3	5
I4	10×10	5	10
I5	10×12	5	10
I6	12×12	5	12
I7	20×20	7	20
I8	20×30	7	20
I9	30×30	7	30

these best solutions produced by ALGN-GA are shown in Table 4, Figure 9, and Figure 10.

It is evident from Table 3 that the value of $P_{\max}$ differs over a set time in each case. The $P_{\max}$ reaches 4 or 3 for the small instances I1-I4. However, for the large instances I5-I9, grids can be nested from the initial to the child grid only once. As table 4 shows, in the best coverage schemes, the strips originate from different grid layers acquired from the ALGN-GA. And the locally nested grids account for more than half of the entire area.

In the best solutions shown in Figure 9, some areas are further subdivided, while others maintain the initial grid. This intuitively suggests that it is unnecessary to use the same grid in the entire area. Some grids in local areas cannot be covered by any strip, such as in the lower left and lower right corners of I5 and the lower left corner of I8, and maintain initial state with no subdivision. Every test in Figure 10 demonstrates a significant increase from generation 0 to generation 100. Each trend exhibits rapid initial growth and a gradual slowing with increasing iterations.

6.3 Experiments of LGN Strategy

6.3.1 Parameter Settings

A series of comparison experiments are conducted for GA and the ALGN-GA. The grid size of GA is fixed at its initial value in each test. Crossover operations in the GA are the same as in the ALGN-GA. No father or child cells are included in the mutation process, so no new strips are constructed. All strips are generated before the initial solution is formed. And the brother strip set S_{ij} (for the initial solution) serves as the source of all solutions through each iteration. In GA, the first layer of each double-layer chromosome is deleted. And the value at each chromosome position represent the strip ID chosen from the corresponding S_{ij}.

Other parameters are set to the same values in both algorithms. And each test is independently repeated 20 times. The GA is tested with a grid layer ranging from $P = 1$ to $P = P_{\max}$, independently. Table 5 compares the average results for the ALGN-GA and GA. Here, "Time" is used to represent the average amount of CPU time in units of seconds.

6.3.2 Comparison Results of GA with the ALGN-GA

Table 5 demonstrates that the average performance of the ALGN-GA is superior than that of GA in 88.9% of instances. For example, in instance I1, GA could not achieve a result in a limited time once $P = 4$, while the ALGN-GA could generate strips in the fourth grid layer. In the nine instances tested, GA reach the corresponding $P_{\max}$ within the limited time for only one instance I2 at the cost of extremely high running time.

Figure 11 shows the iterative processes for

Table 3 ALGN-GA Test Results for Instances I1-I9

Instance	$P_{\max}$	Max(%)	Min(%)	Mean(%)	SD
I1	4	91.6	91.5	91.5	0.000
I2	3	90.7	90.2	90.4	0.001
I3	3	98.2	95.3	97.1	0.010
I4	3	93.2	89.8	92.9	0.002
I5	3	84.6	83.8	84.2	0.003
I6	2	90.6	88.5	89.6	0.007
I7	2	91.8	91.2	91.4	0.002
I8	2	92.5	92.0	92.2	0.002
I9	2	93.8	92.9	93.3	0.003

Table 4 The Strip Layer Distribution and the Ratio of Grids from 1 to $P_{\max}$ for the Best Solutions Produced by ALGN-GA across Instances I1-I9

Instance	Strip number	$P_{\max}$	Number of strips from each layer				Grid proportion (%)			
			1	2	3	4	1	2	3	4
I1	3	4	1	0	0	2	0	0	2.08	97.92
I2	4	3	1	0	3	-	0	0	100	-
I3	5	3	1	2	2	-	0	3	97	-
I4	10	3	3	5	2	-	2.5	3.75	93.75	-
I5	10	3	1	1	8	-	1.25	1.30	97.45	-
I6	12	2	3	9	-	-	1.39	98.61	-	-
I7	20	2	8	12	-	-	9.25	90.75	-	-
I8	20	2	6	14	-	-	8.17	91.83	-	-
I9	20	2	10	20	-	-	11.22	88.78	-	-

Table 5 A Comparison of The ALGN-GA and GA Mean Performance for Instances I1-I9

Instance			I1	I2	I3	I4	I5	I6	I7	I8	I9
ALGN-GA		Mean(%)	**91.5**	90.4	**97.1**	**92.9**	**84.2**	**89.6**	**91.4**	**92.2**	**93.3**
		Time	160.54	17.70	93.73	536.77	981.82	144.91	293.31	548.69	1154.95
		$K_{\max}$	4	3	3	3	3	2	2	2	2
GA	1	Mean(%)	68.2	66.4	93.9	86.2	78.0	85.6	90.4	91.0	92.1
		Time	5.97	8.28	17.94	36.68	34.31	119.95	246.31	597.70	800.24
	2	Mean(%)	89.5	89.0	96.6	92.6	83.9	-	-	-	-
		Time	22.49	56.85	212.66	559.53	630.27	-	-	-	-
	3	Mean(%)	90.8	**92.8**	-	-	-	-	-	-	-
		Time	239.04	833.38	-	-	-	-	-	-	-
	4	Mean(%)	-	-	-	-	-	-	-	-	-
		Time	-	-	-	-	-	-	-	-	-

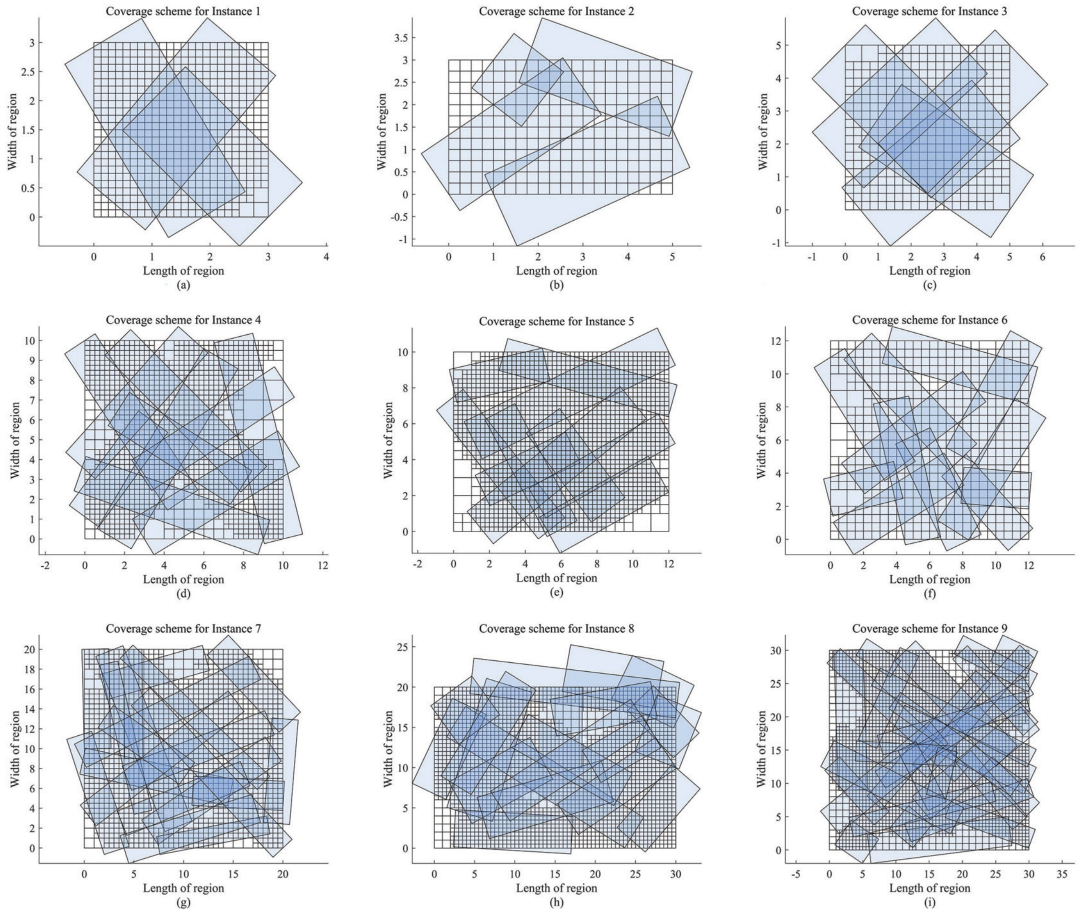


Figure 9 The Best Coverage Schemes Produced by ALGN-GA for Instances I1-I9

the ALGN-GA and GA, where the values do not represent a specific solution but rather the means of 20 independent repeated experiments (making the curves appear smoother). The growth trends for the ALGN-GA and GA are similar, both of which exhibit rapid growth in early iterations and gradual slowing in the later stages. For I1-I5, the GA can operate on at least two layers. As shown, the starting points for the tests of GA are different at different layers. And the starting points for the ALGN-GA iterations are close to the starting points for GA with $P = 1$. However, the results of the ALGN-GA reach or even exceed the final results of GA for $P = 2$ or $P = 3$. For I6-I9, GA and the ALGN-GA with $P = 1$ exhibit nearly the same starting points. However, the final results of ALGN-GA yield significantly higher coverage than GA. In summary, the ALGN-GA achieve obvious improvements over GA and demonstrate the effectiveness of the proposed LGN strategy.

6.4 Comparison with Other Representative Algorithms

Further comparison experiments are conducted with several representative algorithms to demonstrate the superiority of the ALGN-GA. This include the following conventional techniques.

EM: Enumeration Method (EM) is an exact methodology for calculating optimal results.

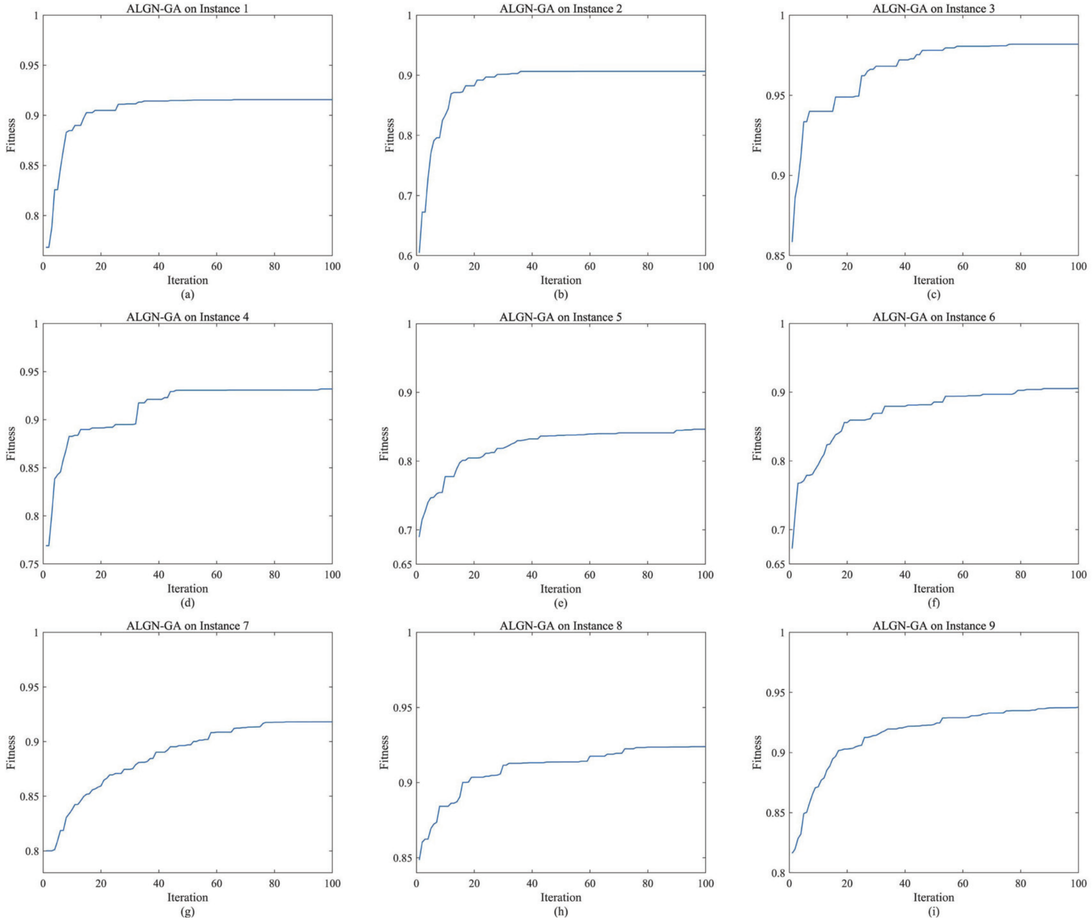


Figure 10 The ALGN-GA Iterative Processes Used to Produce the Best Solutions for Instances I1-I9

In the EM, every possible combination of strips is tried and enumerated individually. The number of enumerated solutions is given by $\prod_{i=1,j=1}^{i=|F|,j=|C_i|} |S_{ij}|$, so it is accurate but extremely time-consuming. And the best EM result forms the exact upper bound of the problem.

ABC (Karaboga and Basturk 2007): Artificial Bee Colony (ABC) is a metaheuristic algorithm inspired by the foraging behavior of honeybees. In the algorithm, the solution is represented as a bee, and a colony of artificial bees is used to search for the optimal solution. Each bee constructs a solution randomly before communicating with other bees to share information about the quality of the solution. The bees then update their solutions based on the information shared by other bees, and the best solutions at the moment are stored in the hive. ABC uses a combination of local search and global search strategies to efficiently explore the search space and find better solutions.

ACO (Wolfe and Sorensen 2000): Ant Colony Optimization (ACO) is a metaheuristic swarm intelligence algorithm inspired by the behavior of real ants in nature. The algorithm is based on the idea that ants communicate with each other by depositing pheromones on the ground, which other ants can follow. In ACO, the solution is represented as an ant and the pheromone is updated according to the value of the objective function corresponding to the solution. By using the principles of

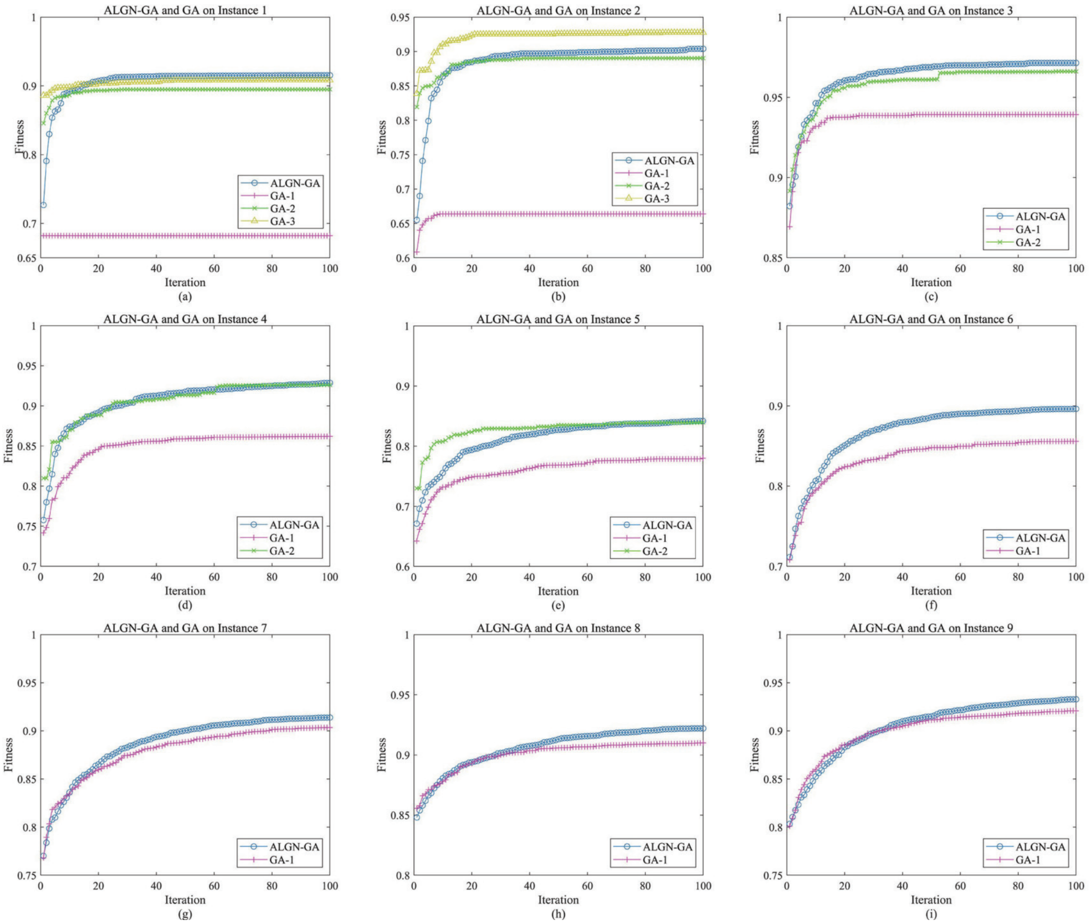


Figure 11 The Average ALGN-GA and GA Iterative Processes for Instances I1-I9

positive feedback and reinforcement, the algorithm iteratively improves the pheromone trail and finds better solutions.

DG (Zhu et al. 2019b): Dynamic Greedy (DG) algorithm tends to fill the space quickly and effectively and identify the available strip with the greatest coverage over remaining uncovered areas. This is a repetitive process of deleting the brother strips of the already used strips, identifying the vacant areas which are not covered by the selected strips, calculating the coverage area of vacant areas for all remaining strips, selecting the strip with the largest coverage and adding it to the solution. This process is repeated until all brother strips are deleted.

TS: Tabu search (TS) algorithm avoids circuitous searches by introducing a storage structure and tabu criteria. Zhu et al. (2019b) applied several algorithms, including TS, and simulated annealing (SA) to the area target EOS observation problem. Experiments demonstrated the TS algorithm is superior to other techniques.

Table 6 shows the average values of solutions determined using the ALGN-GA, GA, LGN-TS, TS, DG, ACO, ABC and EM for 20 repeated tests on nine instances. Table 7 shows the corresponding average running time. The term "GS" represents the grid scale for each instance in various grid layers. The maximum average value for each instance is shown in

Table 6 Average Performance Comparisons for ALGN-GA, GA, LGN-TS, TS, DG, ACO, ABC, and EM across Instances I1-I9

Instance	$P_{\max}$	GL	GS	ALGN -GA(%)	GA(%)	LGN -TS(%)	TS (%)	DG (%)	ACO (%)	ABC (%)	EM (%)
I1	4	1	3×3	**91.5**	68.2	91.4	68.2	67.9	68.2	68.2	68.2
		2	6×6		89.5		89.4	86.2	83.5	82.0	91.3
		3	12×12		90.8		91.0	90.0	85.3	82.5	-
		4	24×24		-		91.3	90.8	-	-	-
I2	3	1	3×5	90.4	66.4	90.3	64.8	66.4	63.4	60.4	69.7
		2	6×10		89.0		90.2	84.5	78.5	80.3	-
		3	12×20		**92.8**		92.4	85.1	80.9	86.9	-
I3	3	1	5×5	**97.1**	93.9	95.6	95.0	91.3	80.6	73.5	96.1
		2	10×10		96.6		95.2	89.2	86.7	80.1	-
		3	20×20		-		-	93.2	-	-	-
I4	3	1	10×10	**92.9**	86.2	89.7	85.6	84.9	71.0	64.4	-
		2	20×20		92.6		89.0	89.6	82.5	76.1	-
		3	40×40		-		-	-	-	-	-
I5	3	1	10×12	**84.2**	78.0	79.8	77.5	75.3	66.1	60.8	-
		2	20×24		83.9		82.5	82.6	79.7	75.5	-
		3	40×48		-		-	-	-	-	-
I6	2	1	12×12	**89.6**	85.6	85.9	82.7	77.8	74.8	58.6	-
		2	24×24		-		84.1	85.5	-	-	-
I7	2	1	20×20	**91.4**	90.4	86.8	86.0	85.7	76.3	55.8	-
		2	40×40		-		83.5	86.0	-	-	-
I8	2	1	20×30	**92.2**	91.0	90.0	86.9	89.7	78.3	54.2	-
		2	40×60		-		-	90.6	-	-	-
I9	2	1	30×30	**93.3**	92.1	86.1	84.3	90.1	74.2	63.0	-
		2	60×60		-		-	-	-	-	-

bold in Table 6.

Tables 6 and 7 demonstrate that the ALGN-GA outperformed other algorithms in 88.9% of instances (excluding I2). The following conclusions can be drawn from these results:

1) For instances with large area areas and initial grid scales, the conventional algorithm can only be applied in the existing grid and cannot generate more strips or expand the search space. However, the ALGN-GA and LGN-TS can be used to divide local areas into smaller grids, producing more strips and better combination optimization results.

2) While the EM can obtain the upper bound of the problem directly, Tables 6 and 7 demonstrate that it is only effective for the low-layer grid in small instances. On the other hand, DG can obtain solutions quickly, but it consistently fails to achieve the best solution available in all instances. This suggests that the local optimization strategy employed by DG may lead to suboptimal solutions due to its limited exploration of the global solution space.

3) In the SMEATO problem, although we discretized the problem by constructing a grid, the number of solution spaces remains enormous, particularly as the area of the target

Table 7 A Comparison of Average Computing Times for ALGN-GA, GA, LGN-TS, TS, DG, ACO, ABC, and EM across Instances I1-I9, with Time Measured in Seconds

Instance	$P_{\max}$	GL	GS	ALGN -GA	GA	LGN -TS	TS	DG	ACO	ABC	EM
I1	4	1	3×3	160.54	5.97	189.23	0.30	0.34	8.23	7.43	0.27
		2	6×6		22.49		0.28	0.36	46.80	22.49	13.90
		3	12×12		239.04		1.91	1.93	522.58	239.36	-
		4	24×24		-		346.09	297.17	-	-	-
I2	3	1	3×5	17.70	8.28	12.91	0.29	0.32	12.29	9.03	0.30
		2	6×10		56.85		0.34	0.43	143.71	64.64	-
		3	12×20		833.38		151.18	16.63	1701.86	819.36	-
I3	3	1	5×5	93.73	17.94	49.39	0.30	0.32	46.43	23.22	515.94
		2	10×10		212.66		0.87	1.19	798.40	210.07	-
		3	20×20		-		-	135.25	-	-	-
I4	3	1	10×10	536.77	36.68	484.24	0.31	0.45	136.66	34.80	-
		2	20×20		630.27		7.00	10.50	2050.32	562.30	-
		3	40×40		-		-	-	-	-	-
I5	3	1	10×12	981.82	34.31	916.07	0.31	0.45	136.66	34.80	-
		2	20×24		630.27		4.55	7.17	2373.86	629.65	-
		3	40×48		-		-	-	-	-	-
I6	2	1	12×12	144.91	119.95	131.85	0.44	1.30	545.55	124.06	-
		2	24×24		-		77.36	91.15	-	-	-
I7	2	1	20×20	293.13	246.31	139.60	1.12	5.52	844.48	253.03	-
		2	40×40		-		289.08	312.92	-	-	-
I8	2	1	20×30	548.69	591.70	501.28	2.41	7.41	1522.06	585.80	-
		2	40×60		-		-	614.68	-	-	-
I9	2	1	30×30	1154.95	800.24	1012.51	17.84	59.03	2082.05	800.24	-
		2	60×60		-		-	-	-	-	-

or the layer of the grid increases, the solution space will increase sharply. ACO and ABC algorithms exhibit similar optimization abilities to GA in cases where the observed targets are small. However, as the area of the target or the layer of the grid increases, their optimization capabilities are significantly lower than those of GA. Despite ensuring that the three algorithms had the same maximum number of function evaluations, ACO often becomes quickly trapped in local optima, and its running time is much higher than that of GA and ABC. The running time of the ABC is comparable to that of GA, but the final solution it finds is not even as good as that of ACO. Among the three population-based algorithms, GA demonstrate significantly better solving abilities than the other two, indicating that GA is more suitable for searching larger solution spaces, supporting the use of GA as the framework for ALGN-GA.

4) The search capability of TS is inferior than that of LGN-TS. In the experiments, LGN-TS searches to a grid layer that TS could not reach in 55.6% of the instances and outperforms TS in 77.8% of instances. This suggests that the proposed LGN strategy is also effective when combined with other algorithm frame-

works.

5) The comparison between ALGN-GA and LGN-TS demonstrates that, for the same instances in the same conditions, average results from ALGN-GA are always superior to those of LGN-TS. However, the LGN-TS running time is generally lower than that of the ALGN-GA. The comparison of TS with GA shows that TS can search strips at a layer that GA cannot (e.g., instances I1, I6, and I7). However, average results from GA are almost always superior to those of TS, which also supports the use of GA as the framework for ALGN-GA.

7. Conclusion

In this study, we investigate the SMEATO problem and discuss the challenges involved in solving the problem. The SMEATO is a continuous space combinatorial optimization problem coupled with computational geometry. It is nearly impossible to develop exact algorithms for finding the global optimal solutions of such a problem. We discretize the area with the strategy of building a grid to reduce the couple with computational geometry. And a local grid nesting strategy is proposed to improve the flexibility of grid division for polygon. On this basis, an ALGN-GA algorithm is designed by adopting a genetic algorithm as a framework for implementing the LGN. In ALGN-GA, an adaptive GA mutation operation is designed for LGN to determine which local regions should be retained in the current grid and which need to be deeply nested. By controlling the nesting of local regions, the proposed ALGN-GA overcome the computational difficulty of expanding the entire area into even grids and achieve a balance between the quality of solutions and the running time. The feasibility and effectiveness of the included LGN strategy are also verified using comparison tests between ALGN-GA and GA, as well as LGN-TS and TS. The performance of the ALGN-GA is compared with GA, EM, DG, ABC, ACO, TS and LGN-TS across nine random instances. The results not only suggest that ALGN-GA offers the best overall performance, they also demonstrate that ALGN-GA is more applicable to the SMEATO at larger scales.

Acknowledgments

The authors express their gratitude to the editors and the anonymous referees for their valuable input, which greatly contributed to enhancing the quality of this paper. This work has been supported in part by the National Natural Science Foundation of China (NSFC), under Grant Nos. 72271074 and 72071064.

Data Availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

References

Arbor A, Holland J (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, USA.

Chen Y, Xu M, Shen X, Zhang G, Lu Z, Xu J (2020). A multi-objective modeling method of multi-satellite imaging task planning for large regional mapping. *Remote Sensing* 12(3): 344.

Dorigo M, Gambardella LM (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1(1): 53-66.

El Garouani A, Mulla DJ, El Garouani S, Knight J (2017). Analysis of urban growth and sprawl from remote sensing data: Case of Fez, Morocco. *International Journal of Sustainable Built Environment* 6(1): 160-169.

Gabrel V, Moulet A, Murat C, Paschos VT (1997). A new single model and derived algorithms for the satellite shot planning problem using graph theory concepts. *Annals of Operations Research* 69(0): 115-134.

Gabrel V, Vanderpooten D (2002). Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research* 139(3): 533-542.

Hu X, Zhu W, An B, Jin P, Xia W (2019). A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem. *Computers & Operations Research* 104: 74-89.

Hu X, Zhu W, Ma H, An B, Zhi Y, Wu Y (2021). Orientational variable-length strip covering problem: A branch-and-price-based algorithm. *European Journal of Operational Research* 289(1): 254-269.

Karaboga D, Basturk B (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization* 39: 459-471.

Karthikeyan L, Chawla I, Mishra AK (2020). A review of remote sensing applications in agriculture for food security: Crop growth and yield, irrigation, and crop losses. *Journal of Hydrology* 586: 124905

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

Li S, Shen X, Yao H, Zhang G, Liu Y (2019). Optimization of lateral swing angles of lunar satellite for region multiple strip imaging task planning. *Geomatics and Information Science of Wuhan University* 44(4): 593-600.

Li X, Zhu J, Mao C (2006). Efficiency optimization of area target observation using earth observation satellites. *Computer Simulation* 12: 24-27.

Luo K (2020). A hybrid binary artificial bee colony algorithm for the satellite photograph scheduling problem. *Engineering Optimization* 52(8): 1421-1440.

Peng G, Wen L, Feng Y, Bai B, Jing Y (2011). Simulated annealing algorithm for EOS scheduling problem with task merging. *Proceedings of 2011 International Conference on Modelling, Identification and Control*. Shanghai, China, June 26-29, 2011.

Perea F, Vazquez R, Galan-Viogue J (2015). Swath-acquisition planning in multiple-satellite missions: An exact and heuristic approach. *IEEE Transactions on Aerospace and Electronic Systems* 51(3): 1717-1725.

Sarkheyli A, Bagheri A, Ghorbani-Vaghei B, Askari-Moghadam R (2013). Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions. *Aerospace Science and Technology* 29(1): 287-295.

Song Y, Zhang Z, Sun K, Yao F, Chen Y (2019). A heuristic genetic algorithm for regional targets' small satellite image downlink scheduling problem. *International Journal of Aerospace Engineering* 2019(PT.1): 1-13.

Srinivas M, Patnaik LM (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 24(4): 656-667.

Sun H, Xia W, Wang Z, Hu X (2021). Agile earth observation satellite scheduling algorithm for emergency tasks based on multiple strategies. *Journal of Systems Science and Systems Engineering* 30(5): 626-646.

Vatti BR (1992). A generic solution to polygon clipping. *Communications of the ACM* 35(7): 56-63.

Wang J, Zhu X, Yang LT, Zhu J, Ma M (2015). Towards dynamic real-time scheduling for multiple earth observation satellites. *Journal of Computer and System Sciences* 81(1): 110-124.

Wang W, Jia D, Xu J, Chu H, Dong X (2020). Review of the development of global marine remote sensing satellite. *Bulletin of Surveying and Mapping* (5): 1.

Wolfe WJ, Sorensen SE (2000). Three scheduling algorithms applied to the earth observing systems domain. *Management Science* 46(1): 148-166.

Yuan Z, He Y, Cai F (2011). Fast algorithm for maneuvering target detection in SAR imagery based on gridding and fusion of texture features. *Geo-spatial Information Science* 14(3): 169-176.

Zhang X, Zhang H, Feng X (2012). The landsat framing algorithm based on WRS-2. *Remote Sensing Information* 27(6): 39-44.

Zhang Z, Zhang N, Feng Z (2014). Multi-satellite control resource scheduling based on ant colony optimization. *Expert Systems with Applications* 41(6): 2816-2823.

Zhou Y, Yan Y, Huang X, Yang Y, Zhang H (2015). Mission planning optimization for the visual inspection of multiple geosynchronous satellites. *Engineering Optimization* 47(11): 1543-1563.

Zhu W, Hu X, Xia W, Jin P (2019a). A two-phase genetic annealing method for integrated earth observation satellite scheduling problems. *Soft Computing* 23: 181-196.

Zhu W, Hu X, Xia W, Sun H (2019b). A three-phase solution method for the scheduling problem of using earth observation satellites to observe polygon requests. *Computers & Industrial Engineering* 130: 97-107.

Ligang Xing is now pursuing his doctoral degree in management science and engineering at Hefei University of Technology. His research interests include the operations research and optimization.

Wei Xia is currently an associate professor in Hefei University of Technology. His research interests include satellite intelligent scheduling and controlling.

Xiaoxuan Hu is currently a professor at Hefei University of Technology. His research interests include satellites scheduling and UAV path planning.

Waiming Zhu is currently working in Hefei University of Technology as a lecturer. His research interests include operations research and satellites scheduling.

Yi Wu is now working for her master degree in management science and engineering at Hefei University of Technology. Her research interests include satellite scheduling.