Two-Archive2 Algorithm for Large-Scale Polygon Targets Observation Scheduling Problem

Xiao-meng Li*

College of Information System and Management, National University of Defense Technology, Changsha, China
*Corresponding author

Keywords: Earth observation, Large-scale polygon targets, Two_Arch2, Inverted generational distance (IGD).

Abstract. In the case of observing a single polygon target for a specified period of time with multiple earth observation satellites, select different observation plans will result in different regional coverage effects. How to schedule the observing scheme, so that the observing activities can achieve the required coverage with less satellite resources, meanwhile minimize the revisit rate of observation area in the shortest time, is a major optimization problem. This paper explores the modeling and solving method of this optimization problem. Firstly, the basic mathematical model is established, including the constraint condition and four optimization objectives. After that, target discretization method is introduced to calculate the regional coverage. Two-Archive2 algorithm (i.e., Two_Arch2) is chosen as the multi-objective evolutionary algorithm to solve this problem. The model and the solution method is used for solving a practical multi-satellite regional observation problem. The validity of the model and the algorithm is verified by the experimental results.

Introduction

Earth observation is the gathering of information about planet Earth’s physical, chemical and biological systems via remote sensing technologies supplemented by earth surveying techniques, encompassing the collection, analysis and presentation of data [1]. With the development of aerospace industry, the number of satellites on orbit continuously increased, earth observation began from a single-satellite observations on point targets evolve to multi-satellite observations of polygon targets.

Earth observation resources are scarce resources because they are costly, the overall cost of R&D, use, maintenance and delivery will be a very large number. In recent years, Earth observation has become technologically increasingly sophisticated. A slight increase in task planning for terrestrial imaging means a significant reduction in cost, with significant economic implications.

It is difficult for satellites to achieve coverage of polygon targets by single imaging due to the size of the field of view (FOV). Therefore, it is necessary to decompose the polygon target before the imaging task planning. By decomposing the polygon target into meta-tasks, meta-tasks are assigned to multi-satellite resources through planning, while, in the form of strips, the distribution of strips over polygon targets is used for realizing the coverage of the whole area.

At present, most of researches on the multi-satellite scheduling are based on the observation of point targets, while the researches based on polygon target observation are still relatively few.

Relevant articles currently include C. Rivett and C. Pontecorvo [2], who examined the problem of observing a polygon target of multiple satellites. They set up a planning model for multi-satellite monitoring of polygon targets and solved them with linear programming software. But only discussed on small-scale polygon targets, there is no solution for large-scale polygon targets observation by multi-satellite coordination.

In this paper, the scheduling problem of large-scale polygon targets under non-agile multi-satellite joint imaging mode is considered. Non-agile satellites decompose the polygon targets by the tracing
method, which requires that the task strips must be parallel to the direction of satellite movement. Obviously, different satellite orbit and orbital cycles will lead to the direction of decomposed strip is inconsistent, resulting in crossover between different strips.

How to decompose the polygon target reasonably to ensure the maximum coverage with the minimum number of satellite resources in the shortest time and reduce the crossover rate of strips is the goal of this scheduling problem. On account of this is a many-objective optimization problem (ManyOPs) which usually refer to those multi-objective problems (MOPs) with more than three objectives, its large numbers of objectives pose challenges to multi-objective evolutionary algorithms (MOEAs) in terms of convergence, diversity, and complexity. Most existing MOEAs can only perform well in one of those three aspects. In view of this, we aim to find a more balanced MOEA on ManyOPs in all three aspects at the same time. Among the existing MOEAs, we found an improved two-archive algorithm (i.e., Two_Arch2) [3] for ManyOPs in this paper, which has a good performance on convergence, diversity, and complexity at the same time. The main idea of this algorithm is to maintain two archives during evolutionary search, where each archive promotes convergence (CA) and diversity (DA) separately. We select this MOEA to solve large-scale polygon targets observation scheduling problem.

The remainder of the paper is organized as follows. We firstly describe the scheduling problem and build the model in section II. In Section III, we will display and evaluate the results of the algorithm. Finally, Section IV gives the conclusion and future work.

Problem Formulation

Table 1. Data symbolic representation.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n_s)</td>
<td>The number of satellites</td>
</tr>
<tr>
<td>(n_b)</td>
<td>The number of orbital cycles in the planning cycle</td>
</tr>
<tr>
<td>(n_k(i,j))</td>
<td>The number of strips in which the satellite i decomposes the target in the jth cycle</td>
</tr>
<tr>
<td>(P)</td>
<td>Discrete point set of polygon targets</td>
</tr>
<tr>
<td>(P_{i,j,k})</td>
<td>The set of discrete points covered by the kth strip of the target decomposition in the jth cycle of the satellite i</td>
</tr>
<tr>
<td>(N(P))</td>
<td>The number of elements in any set P</td>
</tr>
<tr>
<td>(x_{i,j,k})</td>
<td>0-1 variable, indicating whether the kth strip of the target decomposition in the jth rotation cycle of the satellite i is selected to be observed</td>
</tr>
<tr>
<td>(t_{i,j,k})</td>
<td>The starting observation time of the kth strip of the target decomposition in the jth cycle of the satellite i</td>
</tr>
<tr>
<td>(ct_{i,j,k})</td>
<td>The duration observation time of satellite i on the kth strip of the target decomposition in the jth cycle</td>
</tr>
<tr>
<td>(et)</td>
<td>The latest end time of a user-specified observation target</td>
</tr>
</tbody>
</table>

In this paper, the observation task is decomposed into meta-tasks by the tracing method, the main information of a meta-task can be seen as follow:

a. The longitude and latitude coordinates of strip vertexes.
b. The start observation time of strip.

47
c. The start observation side-swing angle of strip.
d. The duration observation time of strip.

Because of the difference between the orbit of different satellites, for the same observation target may decompose into strips with different direction. Meanwhile, owing to the range of the side-swing angle is different, the width of the strip after decomposing is also various. Therefore, the number of meta-tasks generated by the target decomposition is enormous, and the crossover phenomenon between these meta-tasks is also inevitable. How to choose fewer tasks from meta-tasks set to reduce regional repeated observation is the main difficulty of this problem. In the subsections, the problem has been formulated and the symbolic representation is shown in Table 1.

**Constraint Satisfaction Model**

Eq. 1 shows that for the same satellite, the same rotation cycle can only select a strip to observe, which is because non-agile satellites only have side-swing maneuverability. Eq. 2 is the latest end time constraint, it requires that the observation end time for all strips must be less than the observation deadline by the polygon target.

\[
\sum_{k=1}^{n_{s}(i,j)} x_{i,j,k} \leq 1 \quad i = 1, 2, ..., n_s \quad j = 1, 2, ..., n_b
\]

\[
t_{i,j,k} + ct_{i,j,k} \leq et \quad i = 1, 2, ..., n_s \quad j = 1, 2, ..., n_b \quad k = 1, 2, ..., n_s(i,j).
\]

**Objective Function Model**

The four optimization objectives represent minimizing the number of selected strips, minimizing crossover rate, maximizing target coverage rate, and minimizing task completion time.

\[
\min \sum_{i=1}^{n_s} \sum_{j=1}^{n_p} x_{i,j,k}.
\]

\[
\min C\left( \bigcap_{i=1}^{n_s} \bigcap_{j=1}^{n_p} \bigcap_{k=1}^{n_s(i,j)} S\left(P_{i,j,k}\right) \right).
\]

\[
\max C\left( \bigcup_{i=1}^{n_s} \bigcup_{j=1}^{n_p} \bigcup_{k=1}^{n_s(i,j)} S\left(P_{i,j,k}\right) \right).
\]

\[
\min \max \left\{ x_{i,j,k} \left( t_{i,j,k} + ct_{i,j,k} \right) \right\}.
\]

Equation \( S\left(P_{i,j,k}\right) \) and \( C\left(P_{i,j,k}\right) \) can be explained by Eq. 7 and Eq. 8 (\( S\left(P_{i,j,k}\right) \) used for indicating whether a strip is selected, and \( C\left(P_{i,j,k}\right) \) means coverage rate).

\[
S\left(P_{i,j,k}\right) = \begin{cases} P_{i,j,k} & x_{i,j,k} = 1 \\ \emptyset & x_{i,j,k} = 0 \end{cases}
\]

\[
C\left(P_{i,j,k}\right) = \frac{N\left(P_{i,j,k}\right)}{N\left(P\right)}
\]
**Target Discretization**

The discretization of the target is mainly to facilitate the calculation of coverage and crossover rate. It can describe the target through a series of discrete points, and avoid the integration of area, which makes the calculation more convenient and quick.

The discretization process mainly adopts the method of equidistant scattered points, and the precision is controlled by the distance between the discrete points. The more dense the discrete points are, the higher the precision is, but the computational complexity is also increased. Figure 1. gives the schematic diagram of area discretization.

![Figure 1. Schematic diagram of area discretization.](image)

The calculation of the crossover rate \( Cr_{i,j,k} \) of the candidate strip \( P_{i,j,k} \) versus the selected strip set mainly by the following Eq. 9, Where \( p \) is the discrete point, \( P_{c_l} \) is the \( l \) th selected strip, and \( l_c \) is the number of strips.

\[
Cr_{i,j,k} = \frac{N\left(\{ p \mid p \in P_{i,j,k} \land p \in \bigcup_{l=1}^{l_c} P_{c_l}\}\right)}{N}\bigcup_{l=1}^{l_c} P_{c_l} (9)
\]

Figure 2. shows the schematic diagram of cross-coverage, where the blue part represents the cross-coverage area.

![Figure 2. Schematic diagram of cross-coverage area.](image)

**Experimental Results**

**Test Problem**

Two_Arch2 is used for solving large-scale polygon targets observation scheduling problem. In order to verify the applicability and feasibility of the algorithm to solve this practical problem, we adopt a
rectangular large-scale area as the target area to be observed. Strip information includes the latitude and longitude coordinates of vertexes, start observation time and duration observation time, satellite and rotation cycle number. Using three earth observation satellites for imaging and generating multiple meta-tasks by pretreatment. According to the constraints, tasks that do not meet the requirements are deleted and finally 40 strips are selected as input, it means either individual in population has 40 decision variables, their bounds are between 0 and 1, if the strip is selected, its decision variable value takes 1, otherwise the value takes 0. The number of optimization objectives is set to 4, including minimizing the number of selected strips, minimizing crossover rate, maximizing target coverage rate and minimizing task completion time.

**Results and Discussion**

The proposed Two_Arch2 procedure was coded in C# and implemented on a computer with a 1.70 GHz Core (TM) 2 Duo processor with 4.0 GB of RAM memory. In the experiments, we use the same reproduction operators (crossover rate (pc = 1) and mutation rate (pm = 0.1)). The maximum sizes of CA and DA are 30 and 40. Except for test problems with special structures, the Pareto front of a practical problem is usually unknown. Accordingly, the unknown Pareto front is approximated by selecting non-dominated solutions among solution space obtained by run of Two_Arch2 with 100 population size and 500 generations. The approximated Pareto front is shown as four dimension slice graph in Figure 3. All the experiments are repeated for 20 independent runs.

Table 2. shows the mean and standard deviation IGD-metric of obtained solutions in 20 runs. The number of generations ranges from 10 to 100, population sizes in turn are 40, 60, 80, 100. When the population size increased from 80 to 100, the IGD value did not significantly reduce, taking into account the calculation of the problem, select a smaller population size can get relatively satisfactory results. Similarly, when the maximum number of iterations reaches 50, increasing the number of iterations does not result in a significantly lower IGD, consider of computational complexity, it is not necessary to choose higher iteration number.

Figure 4. shows the parallel coordinate plots of the best solution sets obtained by Two_Arch2 with the minimum number of IGD value on the same number of generations. As we can see from the figure, with the increase number of the iterations, the convergence of the solution is gradually become better, although from the point of view of diversity is slightly worse than the results with fewer number of iterations, but also has a good effect.

Figure 3. Four dimension slice graph of approximated PF.
Figure 4. Parallel coordinate plots of the best solution.

Table 2. The mean and standard deviation IGD-metric of obtained solutions in 20 runs.

<table>
<thead>
<tr>
<th>Maxgen</th>
<th>Population size</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>40</td>
<td>0.2043±0.0248</td>
<td>0.2035±0.0223</td>
<td>0.1998±0.0019</td>
<td>0.1991±0.0008</td>
</tr>
<tr>
<td>20</td>
<td>60</td>
<td>0.1527±0.0145</td>
<td>0.1502±0.0065</td>
<td>0.1355±0.0042</td>
<td>0.1328±0.0048</td>
</tr>
<tr>
<td>30</td>
<td>80</td>
<td>0.0989±0.0041</td>
<td>0.0934±0.0052</td>
<td>0.0902±0.0034</td>
<td>0.0897±0.0036</td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td>0.0958±0.0102</td>
<td>0.0926±0.0050</td>
<td>0.0802±0.0043</td>
<td>0.0798±0.0039</td>
</tr>
<tr>
<td>50</td>
<td>40</td>
<td>0.0453±0.0023</td>
<td>0.0358±0.0078</td>
<td>0.0312±0.0125</td>
<td>0.0347±0.0069</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>0.0467±0.0175</td>
<td>0.0328±0.0042</td>
<td>0.0301±0.0112</td>
<td>0.0317±0.0041</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>0.0425±0.0069</td>
<td>0.0349±0.0056</td>
<td>0.0298±0.0073</td>
<td>0.0297±0.0063</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>0.0431±0.0093</td>
<td>0.0333±0.0039</td>
<td>0.0292±0.0025</td>
<td>0.0288±0.0052</td>
</tr>
</tbody>
</table>
Conclusion

In this paper, a large-scale polygon targets observation scheduling problem was proposed, owing to this is a ManyOPs which usually refer to MOPs with more than three objectives, an effective Two-Archive2 algorithm was proposed, which can cope with ManyOPs with satisfactory convergence, diversity, and complexity. The experimental results demonstrate that Two_Arch2 has a good performance for solving this schedule problem. But there are still some shortcomings, firstly, when the problem is modeled, only the constraints of time window and satellite orbit are considered, the limiting factors such as weather, cloud cover, and sensor load are not taken into account. Meanwhile, in the calculation of coverage and crossover rate, due to the use of discrete point division method, resulting in a large amount of calculation. In the process of population evolution also need to repeat the calculation of the objective value. In practical problems, consider of the demand timeliness, this method can not be applied well to the emergency scheduling problem. In the future, the following works can be made. Initially, finding a better method to calculate the coverage rate of the polygon targets is essential, Additionally, it is necessary to find a new multi-objective evolutionary algorithm or a reasonable improvement of the existing algorithm, in order that it can better solve this practical problem.

References


