

## Research Article

# A GRASP for Next Generation Sapphire Image Acquisition Scheduling

Yang Wang,<sup>1</sup> Snezana Mitrovic Minic,<sup>2,3</sup> Robert Leitch,<sup>3</sup> and Abraham P. Punnen<sup>2</sup>

<sup>1</sup>School of Management, Northwestern Polytechnical University, 127 Youyi West Road, Xi'an 710072, China

<sup>2</sup>Department of Mathematics, Simon Fraser University Surrey, 250-13450 102nd Avenue, Surrey, BC, Canada V3T 0A3

<sup>3</sup>MacDonald, Dettwiler and Associates Ltd., Richmond, BC, Canada

Correspondence should be addressed to Yang Wang; [sparkle.wy@gmail.com](mailto:sparkle.wy@gmail.com) and Snezana Mitrovic Minic; [snezanam@sfu.ca](mailto:snezanam@sfu.ca)

Received 25 April 2016; Accepted 10 October 2016

Academic Editor: Paolo Tortora

Copyright © 2016 Yang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates an image acquisition scheduling problem for a Canadian surveillance-of-space satellite named Sapphire that takes images of deep space Earth-orbiting objects. For a set of resident space objects (RSOs) that needs to be imaged within the time horizon of one day, the Sapphire image acquisition scheduling (SIAS) problem is to find a schedule that maximizes the “Figure of Merit” of all the scheduled RSO images. To address the problem, we propose an effective GRASP heuristic that alternates between a randomized greedy constructive procedure and a local search procedure. Experimental comparisons with the currently used greedy algorithm are presented to demonstrate the merit of the proposed algorithm in handling the SIAS problem.

## 1. Introduction

Sapphire is an outward looking satellite used to take images of Earth-orbiting objects in outer space, so-called resident space objects (RSOs). An RSO can be a man-made object or a natural space object. An acquisition request is defined by an RSO that has to be imaged. Within the day, there may be multiple acquisition requests for imaging one RSO. Given a time horizon of one day, the *Sapphire image acquisition scheduling problem* (SAIS) is to find a schedule of a set of acquisition requests that maximizes the sum of the “Figure of Merit” (FoM) of all the scheduled RSO images, where imaging each RSO has time window constraints. The minimal time difference between the two consecutive images of the same RSO is not considered.

The “Figure of Merit” is an integrated measure of five items, including image priority, off-pointing angle, imaging time, manoeuvre angle, and manoeuvre time. The five items are combined as a product in the measure to generally favor images with high priority and small imaging time. The manoeuvre angle is defined as the angle between the pointing vector of the imaging end time for the current RSO and the pointing vector of the imaging start time for the next RSO. The Sapphire manoeuvre time between two acquisition

requests  $i$  and  $j$  calculates the time needed for Sapphire to change its pointing angle from pointing towards RSO <sub>$i$</sub>  at the end of its scheduled imaging time to pointing towards the RSO <sub>$j$</sub>  at the start of its scheduled imaging time. The off-pointing angle measures how much Sapphire is oriented off the most desired position for its solar panels towards the Sun.

The Sapphire image acquisition scheduling problem is a hard discrete optimization problem, which can be modeled as a prize collecting selective asymmetric traveling salesman problem with multiple time windows for each customer, with service times, and with time-dependent distances between the customers. Each imaging request can be represented by a node of a directed graph  $G$ . A directed arc is added between any two imaging requests that can be taken one after another. The weight of an arc represents the duration of the Sapphire maneuver needed for imaging the two corresponding RSOs. Each node has a priority (prize) and the service time that corresponds to the imaging time.

The daily photograph scheduling of the SPOT5 satellite for specified areas on the Earth's surface has been widely studied in the literature. Bensana et al. (1996) [1] formulated the daily management of an Earth observation satellite as a variable valued constraint satisfaction problem, which is

handled by exact and approximate methods. Vasquez and Hao (2001) [2] formulated the daily photograph scheduling of an Earth observation satellite as a knapsack model and proposed a tabu search algorithm to determine the scheduling. Wolfe and Sorensen (2000) [3] defined the scheduling problem as a time window constrained packing problem and devised a fast and simple priority dispatch method, a look ahead algorithm, and a genetic algorithm. Mansour and Dessouky (2010) [4] presented a genetic algorithm to maximize a multicriteria objective including the profit and the number of acquired photographs. Ribeiro et al. (2010) [5] employed a commercial solver combined with a strengthened formulation based on valid inequalities arising from the node packing and 3-regular independence system polyhedra.

The photograph acquisition scheduling problems for an agile Earth observation satellite or for a set of agile Earth observation satellites have been solved by varying approaches. Lemaître et al. (2002) [6] developed four different methods to solve the track selection and scheduling problem for agile Earth observation satellites, including a fast greedy algorithm, a dynamic programming algorithm, a constraint programming algorithm, and a local search algorithm. Bianchessi et al. (2007) [7] considered the problem of selecting and scheduling requests for a constellation of agile satellites and devised a tabu search algorithm with solution quality evaluated by a column generation method. Habet et al. (2010) [8] addressed an agile Earth observing satellite scheduling problem with stereoscopic and time window visibility constraints by hybridizing a tabu search and a systemic search that uses partial enumerations. Upper bounds are given by a dynamic programming algorithm on a relaxed problem. Jang et al. (2013) [9] studied an image collection planning problem for a Korean satellite KOMPSAT-2 and developed a binary integer programming model and a heuristic solution procedure using the Lagrangian relaxation and subgradient methods. Tangpattanakul et al. (2015) [10] modeled the selection and scheduling of observations taken by an agile satellite as a two-objective optimization problem and solved it with an indicator-based multiobjective local search. Xu et al. (2016) [11] investigated an Earth observation scheduling problem for agile satellites under a time window constraint and resource constraints, which are solved by several constructive procedures including an ant colony optimization approach and two heuristics.

Currently used Sapphire image acquisition scheduling algorithm is a greedy heuristic (GH). In this paper, we explore a GRASP heuristic (greedy randomized adaptive search procedure) that integrates a randomized greedy constructive procedure and a simple insertion improvement procedure to enhance the scheduling performance. Our experimental study has demonstrated that the developed algorithm is not only able to improve the objective “Figure of Merit,” but also improves the number of scheduled requests as a bonus.

The paper is organized as follows. In Section 2, we present the problem model of the Sapphire image acquisition scheduling problem. Section 3 presents the proposed GRASP algorithm. Section 4 provides the results of the computational experiments. Finally, concluding remarks are presented in Section 5.

## 2. Problem Model

The Sapphire image acquisition scheduling (SIAS) problem is to find a schedule that maximizes the sum of the “Figure of Merit” of all the scheduling requests. A solution  $S$  to the SIAS problem, a schedule, is a set  $S \in R$  of the scheduled requests, where  $R$  represents all the acquisition requests. The model of the SIAS problem is formulated as

$$\begin{aligned} \text{Maximize} \quad & f(S) = \sum_{i \in S} \text{FoM}_i \\ \text{Subject to} \quad & S \in \Lambda, \end{aligned} \quad (1)$$

where the set  $\Lambda$  contains all schedules  $S$  satisfying the following constraints:

- (1) Within the time horizon of one day, each RSO can be observed in multiple different time windows
- (2) Each time the current request  $i$  to image RSO $_i$  is finished, Sapphire needs manoeuvre time to take the next request  $j$  to image RSO $_j$ , and the minimum manoeuvre time is 80 seconds and increases as a function of the manoeuvre angle magnitude
- (3) The imaging time for each RSO is known
- (4) The next request cannot start earlier than the end time of the current request and must be finished by its time window
- (5) No imaging activity happens outside the planning time horizon

We assume the following:

- (1) Memory on-board is not a constraint
- (2) The energy for moving or maneuvering of the satellite is not an issue (the payload is rigidly attached to the satellite)
- (3) Collecting the solar energy is not a task that needs to be scheduled; however, the satellite solar panel orientation relative to the Sun is part of the “Figure of Merit” for the RSO selection
- (4) Sapphire telescope is focused at infinity and this is a constant

## 3. GRASP: Greedy Randomized Adaptive Search Procedures

GRASP, which stands for Greedy Randomized Adaptive Search Procedure, is an effective approach to solving many optimization problems [12, 13]. It mainly consists of a randomized greedy component to produce an initial solution and a solution improvement component to refine the initial solution. The randomized greedy component integrates greedy and random features to sample solutions in the solution space so as to produce initial solutions with good solution quality and diversity. The solution improvement component can be any local search based heuristics, for example, tabu search [14, 15] and variable neighborhood search [16, 17].

```

(1) start = 0,  $f^* = 0$ 
(2) while start <  $\delta$  do
(3)   if start = 0 then
(4)     Apply a greedy heuristic to produce an initial scheduling  $S^0$ 
(5)   else
(6)     Apply a randomized greedy heuristic to produce an initial scheduling  $S^0$ 
(7)   end if
(8)   Apply a simple insertion improvement heuristic to get an improved scheduling  $S$ 
(9)    $f = \sum_{i|i \in S} \text{FoM}_i$ 
(10)  if  $f > f^*$  then
(11)     $f^* = f, S^* = S$ 
(12)  end if
(13)  start = start + 1
(14) end while
(15) Return the best scheduling  $S^*$  and  $f^*$ 

```

ALGORITHM 1: Pseudocode of GRASP for the SIAS problem.

```

(1) Let  $Q$  denote the set of the RSO imaging requests,  $T$  the duration of the planning horizon with the default setting of 24 hours,  $\bar{m}$  the minimal manoeuvre time with the default setting of 80 seconds and  $d$  the time step for each constructive iteration
(2)  $t = 0$ 
(3) while  $t + \bar{m} < T$  or  $Q \neq \phi$  do
(4)   Find all RSOs in  $Q$  that are visible in time interval  $[t + \bar{m}, t + \bar{m} + d]$  and put them in set  $S$ 
(5)   if  $S = \phi$  then
(6)      $t = t + d$ 
(7)   else
(8)     Calculate the FoM value for each RSO in  $S$ 
(9)     Identify cls RSOs from  $S$  with largest FoM values and randomly select one from them as  $\text{RSO}_b$ 
(10)    Label the start of the imaging time by  $t_s$  and the end of the imaging time by  $t_e$  for  $\text{RSO}_b$ 
(11)    Put  $\text{RSO}_b$  at the end of the schedule and remove it from  $Q$ 
(12)    Let  $t = t_e$ 
(13)  end if
(14) end while

```

ALGORITHM 2: Pseudocode of the randomized greedy heuristic.

Our proposed GRASP for the SIAS problem is shown in Algorithm 1. Since the purpose of this work is to improve performance of the previous greedy heuristic, our algorithm begins by generating an initial schedule using this greedy heuristic. If a “start” of this multistart algorithm finds a schedule with a better objective value  $f$  than the previously best found value  $f^*$ , the algorithm updates the best schedule  $S^*$  and the objective value  $f^*$ . After  $\delta$  multistarts, that is, after the “start” variable reaches  $\delta$  ( $\delta$  is a parameter), the GRASP is terminated. The following sections present details of the GRASP adaptation to the SIAS problem.

**3.1. The Randomized Greedy Heuristic.** The randomized greedy heuristic (RGH) generates a schedule by appending the imaging requests one by one at the end of the previous schedule. Each iteration in RGH considers such a set  $ST$  of imaging requests for the RSOs that are visible for the next 600 seconds from the imaging end time of the last scheduled request. Then, it calculates the FoM value for each request

in the set  $ST$ . Finally, it creates a restricted candidate set of cls imaging requests with maximum FoM values that can be feasibly added at the end of the schedule, from which an imaging request is randomly chosen to insert in the schedule. The RGH repeats the above-mentioned iterations until there is not enough time for scheduling one more imaging request within the given planning horizon or all the imaging requests are scheduled. Note that the complexity of each RGH iteration is bounded by  $O(q)$ , where  $q$  denotes the total number of RSO imaging requests. Algorithm 2 gives the pseudocode of the proposed randomized greedy heuristic.

**3.2. The Simple Insertion Improvement Heuristic**

**3.2.1. Move Operators.** We propose a Simple Insertion Improvement Heuristic (SIIH) for the local search procedure in GRASP, which combines the swap move, the backward move, and the insertion move, to collectively conduct neighborhood exploration. The three move operators are defined as follows.

- (i) *Swap Move*. Given a feasible schedule, the swap move replaces a scheduled request  $i$  with such an unscheduled request  $j$  that satisfies the following: (1)  $\text{RSO}_j$  is visible between the time intervals  $[t_e(i-1)+80, t_s(i+1)]$ ; (2)  $\text{RSO}_j$  is permitted to be imaged not later than the time  $t_s(i+1) - \text{ImgTime}(j)$ .  $\text{RSO}_j$  is permitted to be imaged if the total time for the manoeuvre time  $\text{mt}(i-1, j)$  from the precedent  $\text{RSO}_{i-1}$  to  $\text{RSO}_j$ , the manoeuvre time  $\text{mt}(j, i+1)$  from  $\text{RSO}_j$  to the next  $\text{RSO}_{i+1}$ , and the imaging time  $\text{ImgTime}(j)$  for the  $\text{RSO}_j$  are no more than  $t_s(i+1) - t_e(i-1)$ ; that is,  $t_s(i+1) - t_e(i-1) \geq \text{mt}(i-1, j) + \text{ImgTime}(j) + \text{mt}(j, i+1)$ .
- (ii) *Backward Move*. The backward move enables a sequence of requests that follows the newly inserted request to be imaged at an earlier time, which thus results in an available time span. For a request  $k$ , its imaging time can be shifted to be an earlier time if it satisfies the following: (1)  $\text{RSO}_k$  is visible between the time intervals  $[t_e(k-1)+80, t_s(k)]$ ; (2) the manoeuvre time from the precedent  $\text{RSO}_{k-1}$  to the request  $\text{RSO}_k$  is no more than  $t_s(k) - t_e(k-1)$ ; that is,  $\text{mt}(k-1, k) < t_s(k) - t_e(k-1)$ ; (3) the next  $\text{RSO}_{k+1}$  can be imaged not later than the time  $t_s(k+1)$ .
- (iii) *Insertion Move*. The insertion move inserts an unscheduled request into the current schedule when a sufficient time span occurs after a sequence of backward moves. An unscheduled request  $i$  is qualified to be inserted behind the request  $j$  that performs the last backward move if the following conditions are satisfied: (1)  $\text{RSO}_i$  is visible between the time intervals  $[t_e(j) + 80, t_s(j+1)]$ ; (2)  $\text{RSO}_i$  can be imaged not later than the time  $t_s(j+1) - \text{ImgTime}(i)$ ; that is,  $t_s(j+1) - t_e(j) \geq \text{mt}(j, i) + \text{ImgTime}(i) + \text{mt}(i, j+1)$ .

The complexity to determine the visibility of a RSO during a given time interval, the manoeuvre time between two consecutive RSO imaging requests, and the qualification of imaging an RSO request during a given time interval are all  $O(1)$ . Hence, both the complexity of performing a swap move and an insertion move are  $O(1)$  while the complexity of performing a backward move is bounded by  $O(q)$ .

**3.2.2. Simple Insertion Improvement Heuristic.** For a feasible schedule obtained from the randomized greedy procedure, the SIIH iteratively searches for a better schedule based on the swap move, the backward move, and the insertion move. The swap moves are composed of making swaps between scheduled requests and unscheduled requests. Due to a large number of the swap moves and an expensive time consumption for calculating the objective value for each swap move, the SIIH employs a first-improvement strategy rather than a best-improvement strategy. Precisely, each iteration in the SIIH performs a swap move as long as the resulting schedule after the performance of this move gets a better objective value. Considering that the order to examine the swap moves could affect the search efficiency under the first-improvement strategy, the SIIH examines the scheduled requests sequentially and the unscheduled requests in the

order of their priority. This is derived from the observation that a high-priority request has a higher chance to get a large FoM value, which results in a better schedule.

Each iteration in the SIIH conducts the following operations. For each scheduled request  $i$ , all unscheduled requests are first evaluated for the qualification to be involved in a swap move. If such a swap move  $(i, j)$  is found and a better objective value of the resulting schedule after performing this swap move is obtained, the following operations are conducted: (1) this unscheduled request  $j$  is inserted into the schedule by replacing the scheduled request  $i$ ; (2) the backward move operates to get a continuous and long time span by making the requests behind the just inserted request  $j$  start at an earlier time if possible; (3) as long as the starting time of some requests gets earlier, an insertion move is attempted if an unscheduled request is qualified to be inserted into the newly available time span. If no better objective value is found by performing this swap move, then the swap move is not performed and the next unscheduled request is examined. Each time a swap move is performed, the search sequentially goes to the next scheduled request and repeats the above-mentioned operations. Algorithm 3 shows the pseudocode of the simple insertion improvement heuristic, which terminates when no swap move is able to produce a schedule with a better objective value. The complexity of each SIIH iteration is bounded by  $O(q^2)$ .

## 4. Computational Results

In this section, we compare the proposed GRASP algorithm with the greedy heuristic (GH) currently used in the Sapphire image acquisition scheduling project. The GH works similarly as the randomized greedy heuristic shown in Algorithm 2, but it requires that each constructive iteration selects an imaging request with the best FoM value. The experiments are evaluated on 10 problem instances, each of which is generated from a model in Satellite Toolkit (STK) using an actual Sapphire Task List containing RSO orbit Two Line Elements (TLEs). The content of the Sapphire Task List used is a close representative of the real-world Sapphire Task List. Each problem instance includes 630 RSO acquisition requests and records a time series of pointing vectors covering a 24-hour period.

The algorithms are programmed in C++ and tested on a PC with i7-2600 3.4 GHz CPU and 16 GB memory. Given the stochastic nature of the GRASP algorithm, each instance is run 10 times with different random seeds. The parameters in the experiments are set as follows: the candidate list size  $\text{cls} = 3$  and the maximum starts  $\delta = 100$ .

Table 1 presents the best results of GH and GRASP in terms of the number of imaged requests  $\text{ImgReq}^{\#}$ , the objective value FoM, the improvement percentage of GRASP over GH (Imp%), and the running (computational) time in seconds (Time). The last row summarizes the average improvement of GRASP over GH. From Table 1, we observe that GRASP is able to improve the objective values FoM obtained by GH for 34.46% on average. Furthermore, in 9 out of 10 instances GRASP was able to schedule more imaging requests than GH, regardless to the fact that

```

(1) Input: a given schedule  $S$  and its objective value  $f$ 
(2) Output: the refined schedule  $S^*$  and its objective value  $f^*$ 
(3)  $best\_fg = 0$ 
(4) repeat
(5)    $iter\_fg = 0$ 
(6)   for each scheduled request  $i$  do
(7)     Sort the unscheduled requests according to the priority in a non-increasing order
(8)     Identify the first such unscheduled request  $j$  that being swapped with the scheduled request  $i$  leads to a feasible schedule and gets a better objective value  $f$ 
(9)     if such a request  $j$  exists then
(10)      Swap the request  $j$  with the request  $i$  and compute the corresponding objective gain  $fg$ 
(11)      Conduct feasible backward moves for each request following  $i$ , and update the corresponding objective gain  $fg$ 
(12)      Conduct a request insertion operation between the end time of the request performing the last backward move and the start time of its next request, and update the corresponding objective gain  $fg$ 
(13)       $iter\_fg = iter\_fg + fg$ 
(14)     end if
(15)     if  $iter\_fg > best\_fg$  then
(16)        $best\_fg = iter\_fg, f^* = f + best\_fg$ 
(17)       Record the current schedule as the best schedule  $S^*$ 
(18)     end if
(19)   end for
(20)    $f = f + iter\_fg$ 
(21) until  $iter\_fg > 0$ 

```

ALGORITHM 3: Pseudocode of the Simple Insertion Improvement Heuristic.

TABLE 1: Best results of the GH versus GRASP.

Instances	GH			GRASP			
	ImgReq <sup>#</sup>	FoM	Time (s)	ImgReq <sup>#</sup> (Imp%)	FoM	Imp%	Time (s)
Day 1	488.0	0.032758	5	514 (5.33%)	0.051292	56.58%	1400
Day 2	551.0	0.028856	9	562 (2.00%)	0.031378	8.74%	1440
Day 3	572.0	0.027112	9	572 (0.00%)	0.036878	36.02%	1360
Day 4	559.0	0.025122	9	581 (3.94%)	0.030325	20.71%	1340
Day 5	579.0	0.025286	9	584 (0.86%)	0.034333	35.78%	1360
Day 6	567.0	0.026977	9	581 (2.47%)	0.027701	2.68%	1220
Day 7	568.0	0.025356	9	588 (3.52%)	0.033747	33.09%	1340
Day 8	580.0	0.023662	9	589 (1.55%)	0.032998	39.46%	1290
Day 9	576.0	0.021414	9	583 (1.22%)	0.028848	34.72%	1270
Day 10	570.0	0.019205	9	590 (3.51%)	0.033970	76.88%	1280
Avg.						34.46%	

the number of imaged requests  $ImgReq^{\#}$  was not the part of the objective function that the algorithm was optimizing. However, this remarkable performance of GRASP consumes more computational time, which is mainly attributed to the time-consuming Simple Insertion Improvement Heuristic.

Table 2 presents the average results of GH and GRASP with the similar statistics given as in Table 1. As shown in Table 2, GRASP is able to achieve improved results in terms of FoM for all the instances with an average improvement of 23.97%. In terms of  $ImgReq^{\#}$ , GRASP performs better than GH for all the instances except the 3 instances (Day 3, Day 5, and Day 9). When further comparing with the best results shown in Table 1, we find that for each instance the average  $ImgReq^{\#}$  and FoM obtained by GRASP are inferior

to their best counterparts, but the average computational time is much shorter. This observation indicates that the proposed GRASP algorithm could be further improved by increasing its robustness.

It is noteworthy that the “Figure of Merit” is always increasing but the number of images captured not necessarily increases. On the one hand, the objective function of the SIAS problem is the summation of the “Figure of Merit” of all the scheduled images. Thus, a better schedule is closely related to the number of the scheduled images. On the other hand, the “Figure of Merit” also considers the image priority. Although capturing high-priority images contributes to larger “Figure of Merit,” they are likely to take more imaging time and result in a reduced number of total scheduled images during the given time horizon.

TABLE 2: Average results of the GH versus GRASP.

Instances	GH			GRASP			
	ImgReq <sup>#</sup>	FoM	Time (s)	ImgReq <sup>#</sup>	FoM	Imp%	Time (s)
Day 1	488.0	0.032758	5	493.4	0.048963	49.47%	140
Day 2	551.0	0.028856	9	553.0	0.029121	0.92%	144
Day 3	572.0	0.027112	9	563.9	0.028773	6.13%	136
Day 4	559.0	0.025122	9	571.9	0.028059	11.69%	134
Day 5	579.0	0.025286	9	572.8	0.032134	27.08%	136
Day 6	567.0	0.026977	9	569.3	0.027049	0.27%	122
Day 7	568.0	0.025356	9	574.9	0.031408	23.87%	134
Day 8	580.0	0.023662	9	580.7	0.030054	27.01%	129
Day 9	576.0	0.021414	9	573.9	0.027035	26.25%	127
Day 10	570.0	0.019205	9	579.2	0.032074	67.01%	128
Avg.						23.97%	

In summary, the experiments demonstrate the effectiveness of the proposed GRASP algorithm for solving the Sapphire image acquisition scheduling problem.

## 5. Conclusion

We developed a GRASP algorithm for solving the Sapphire image acquisition scheduling problem with time window constraints. The proposed algorithm integrates a randomized greedy procedure and a local search procedure to improve the scheduling performance. The local search procedure employs a collection of swap, backward, and insertion moves to cooperatively conduct the neighborhood exploration. Experimental comparisons with the currently used greedy algorithm disclose that the proposed algorithm is able to obtain improvements in terms of the optimized objective and the number of imaged requests. Potential future work can be considered for robustness improvement of the proposed GRASP algorithm.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This research work was supported by an NSERC CRD grant awarded to Abraham P. Punnen with MacDonald, Dettwiler and Associates Ltd. as industrial collaborator, National Natural Science Foundation of China (Grant No. 71501157), Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2016JQ7007), and China Postdoctoral Science Foundation (Grant No. 2015M580873) awarded to Yang Wang.

## References

- [1] E. Bensana, G. Verfaillie, J. C. Agnse, N. Bataille, and D. Blumstein, "Exact and inexact methods for the daily management of an earth observation satellite," *Space Mission Operations and Ground Data SystemsSpaceOps*, vol. 96, no. 394, pp. 507–514, 1996.
- [2] M. Vasquez and J.-K. Hao, "A 'logic-constrained' knapsack formulation and a tabu algorithm for the daily photograph scheduling of an Earth observation satellite," *Computational Optimization and Applications*, vol. 20, no. 2, pp. 137–157, 2001.
- [3] W. J. Wolfe and S. E. Sorensen, "Three scheduling algorithms applied to the earth observing systems domain," *Management Science*, vol. 46, no. 1, pp. 148–168, 2000.
- [4] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite," *Computers and Industrial Engineering*, vol. 58, no. 3, pp. 509–520, 2010.
- [5] G. M. Ribeiro, M. F. Constantino, and L. A. N. Lorena, "Strong formulation for the SPOT 5 daily photograph scheduling problem," *Journal of Combinatorial Optimization*, vol. 20, no. 4, pp. 385–398, 2010.
- [6] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, 2002.
- [7] 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.
- [8] 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, no. 2, pp. 307–333, 2010.
- [9] J. Jang, J. Choi, H.-J. Bae, and I.-C. Choi, "Image collection planning for Korea multi-purpose SATellite-2," *European Journal of Operational Research*, vol. 230, no. 1, pp. 190–199, 2013.
- [10] P. Tangpattanakul, N. Jozefowicz, 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.
- [11] R. Xu, H. P. Chen, X. L. Liang, and H. M. 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.
- [12] M. G. Resende and C. C. Ribeiro, "Greedy randomized adaptive search procedures," in *Handbook of metaheuristics*, vol. 57, pp. 219–249, Kluwer Academic, Boston, Mass, USA, 2003.

- [13] Y. Wang, Z. Lü, F. Glover, and J.-K. Hao, "Probabilistic GRASP-Tabu search algorithms for the UBQP problem," *Computers & Operations Research*, vol. 40, no. 12, pp. 3100–3107, 2013.
- [14] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [15] Z. Lü and J.-K. Hao, "Adaptive neighborhood search for nurse rostering," *European Journal of Operational Research*, vol. 218, no. 3, pp. 865–876, 2012.
- [16] P. Hansen and N. Mladenović, "Variable neighborhood search: principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.
- [17] Q. Wu, J.-K. Hao, and F. Glover, "Multi-neighborhood tabu search for the maximum weight clique problem," *Annals of Operations Research*, vol. 196, pp. 611–634, 2012.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

