HAL
open science

A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite

Panwadee Tangpattanakul, Nicolas Jozefowiez, Pierre Lopez

To cite this version:

Panwadee Tangpattanakul, Nicolas Jozefowiez, Pierre Lopez. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. European Journal of Operational Research, 2015, 245 (2), pp.542-554. 10.1016/j.ejor.2015.03.011 . hal-01162839

HAL Id: hal-01162839

https://hal.science/hal-01162839v1

Submitted on 11 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite

Panwadee Tangpattanakul

Geo-Informatics and Space Technology Development Agency (GISTDA), 120 The Government Complex, Chaeng Wattana Road, Lak Si, Bangkok 10210, Thailand

Nicolas Jozefowiez

CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

Pierre Lopez

CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, LAAS, F-31400 Toulouse, France

Abstract

This paper presents an indicator-based multi-objective local search (IBMOLS) to solve a multi-objective optimization problem. The problem concerns the selection and scheduling of observations for an agile Earth observing satellite. The mission of an Earth observing satellite is to obtain photographs of the Earth surface to satisfy user requirements. Requests from several users have to be managed before transmitting an order, which is a sequence of selected acquisitions, to the satellite. The obtained sequence has to optimize two objectives under operation constraints. The objectives are to maximize the total profit of the selected acquisitions and simultaneously to ensure the fairness of resource sharing by minimizing the maximum profit difference between users. Experiments are conducted on realistic instances. Hypervolumes of the approximate Pareto fronts are computed and the results from IBMOLS are compared with the results from the biased random-key genetic algorithm (BRKGA).

Keywords: Multi-objective optimization, Earth observing satellite, scheduling, local search

1. Introduction and Background

This paper addresses a multi-objective optimization problem associated with selecting and scheduling observations of an agile Earth observing satellite. We consider the case where multiple users order requests to the satellite. A local search is proposed to solve the problem and experiments are conducted on realistic instances.

The mission of Earth observing satellites (EOSs) is to obtain photographs of the Earth surface, in order to satisfy the requirements from users. EOSs can acquire photographs, while moving along their orbits. They spend a period of several days to perform a cycle of orbit. The whole area of the Earth is viewed, when the satellites complete a full cycle [13]. EOSs carry different instruments depending on their usages, e.g. optical camera or infrared camera. Most of them operate at low altitudes. Hence, when they move over the visible areas of the required photographs, the photographs can be captured as in Figure 1. Then, the satellites will try to transfer the data of the acquired images directly to the ground station center after acquiring them, if possible. Otherwise, the data are stored in the on-board limited memory until the satellites are in the possible transferring range to the ground station center.

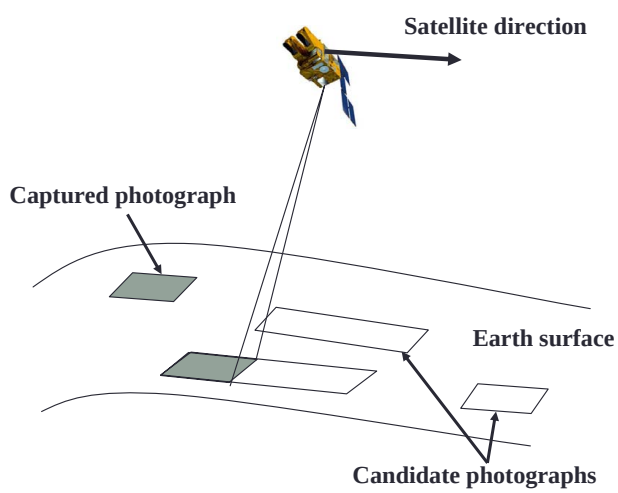

Figure 1: The satellite captures the photographs [23].

Among the various types of EOSs, only so-called "agile" satellites are considered in this paper. An agile EOS is equipped with only one fixed on-board

camera, but the satellite uses an attitude and orbit control system (AOCS) to be able to turn around three axes: roll, pitch, and yaw [18]. An example of an agile satellite is PLEIADES, which was developed by the CNES, the French Space Agency. The starting time for taking each image of this satellite is not fixed, but it must be in a given time interval, which is called a time window. Therefore, an agile satellite has an important advantage when compared to a non-agile satellite. On the one hand, this gives agile satellite better efficiency of the whole system. On the other hand, the problem of selecting and scheduling the candidate images is more difficult to solve, since the search space under consideration is larger [19].

In this work, the satellite management process is considered when several users order requests to a ground station center. The requests cannot be assigned directly to a satellite; the ground station center has to select and schedule the candidate images, according to some limitations of the satellite, before the obtained sequence is transmitted.

For solving the Earth observation scheduling problem, there are several studies on agile EOSs. For example, a combination of genetic algorithm and simulated annealing was proposed to solve this problem in [21]. The performance of the proposed algorithm was compared with the simulated annealing alone. In [19], four methods consisting of a greedy algorithm, a dynamic programming procedure, a constraint programming model, and a local search method were applied in order to solve a simplified version of the scheduling problem for agile EOSs.

The ROADEF 2003 challenge was about the management problem of an agile EOS mission (see `http://challenge.roadef.org/2003/en/`). The challenge aims at finding a feasible schedule that maximizes the total profit, computed from the sum of request gains, which are associated with the complete or partial acquisition of each request. All the data description and optimization criterion are explained in [29]. Note that the problem considered in this challenge was a simplified version of the real Earth observation satellite management problem; for example, neither data download nor energy and thermal limitations are taken into account.

The winner of this challenge used an algorithm based on simulated annealing for solving the scheduling problem [16]. The second prize winner proposed an algorithm based on tabu search [8]. The authors adapted the unified tabu search algorithm [7], which was developed for the vehicle routing problem with time windows. Moreover, a tabu search algorithm hybridized with a systematic search was applied to solve this problem in [13]. All these works

considered the scheduling problem for an agile EOS as a mono-objective optimization problem (total profit maximization).

Our work considers the acquisition scheduling problem of an agile EOS, where the requests emanate from several different users. We need to optimize two objective functions, which are to maximize a total profit and simultaneously ensure the fairness of resource sharing for all users. Thus, this problem is modeled as a multi-objective optimization problem. The second objective, which is added in order to ensure the fairness, amounts to minimize the maximum profit difference between users. Some researchers studied multi-objective optimization problems for space applications [1][10][30]. Furthermore, some literature considered as an objective the fairness among users [20]. Multiple end-users of agile EOSs were considered and sharing principles were adopted to select the subset of candidates based on utility levels. In [3] and [17], the use of two objective functions related to fairness and efficiency was proposed. Three ways were discussed for solving this sharing problem: the first one gives priority to fairness, the second one to efficiency, and the third one computes a set of trade-offs to help a human to make decisions. For the multicriteria methods, instead of building a complete set of nondominated solutions, the authors only searched for a decision close to the line with a specified slope on the objective function plane. In [6], the selecting and scheduling requests for the multi-satellite, multi-orbit, and multi-user were studied, and tabu search was used to solve the problem. The fairness was taken into account, but it was not considered as an objective function. The authors borrowed an ordered weighted average from [31] to ensure the fairness of the solutions. The experiments test these algorithms with the data instances provided by the CNES.

This paper proposes an indicator-based multi-objective local search, which is a multi-objective metaheuristic algorithm, for selecting and scheduling the subset of candidate photographs. Section 2 presents the description of the multi-user Earth observation scheduling problem. Then, the indicator-based multi-objective local search is explained in Section 3. Section 4 presents the computational results. This section compares the results from the indicator-based multi-objective local search and from the biased random-key genetic algorithm. Finally, conclusions and perspectives are discussed in Section 5.

2. Problem description

2.1. Informal presentation

The instances, which are modified from the ROADEF 2003 challenge instances, will be described in detail. They will be used in the experiments, for testing the performance of the proposed algorithm in our work.

Each request can be of two types: mono or stereo. Each area is taken only once for mono requests, whereas for stereo requests, each area must be acquired twice in the same direction but from different angles. Two possible shapes of request, which are a spot or a polygon, can be required. The spot is a small circular area with a radius of less than 10 km. The polygon is a polygonal area ranging from 20 to 100 km. Both shapes have to be managed by transforming the requests into several rectangular shapes called strips. Each polygon is decomposed into several strips of the same width but with variable lengths. A spot is considered as a single strip. Each strip can be taken once at a time by the camera on the satellite. An example of request shapes and order for taking the strips after management is illustrated in Figure 2. There are two possible directions to acquire each strip. Both directions are parallel to the length of the strip, but in the opposite directions as shown in Figure 3. Among two of them, only one acquired direction can be selected. The strip, associated with one possible acquired direction, is called an acquisition. Thus, each strip consists of two possible acquisitions. The interval of possible starting times for taking each acquisition can be computed, depending on the acquired direction, from the earliest and latest visible time of the two extremities of the strip, and the acquired duration time of the strip.

Each acquisition generates a profit. Thus, for the observation scheduling problem, the objective is total profit maximization. The total profit is computed depending on the acquired area of each request. The profit of each acquired request can be computed by using a piecewise linear function of gain. This function is associated with a fraction of the acquired useful area over the whole area of each request, as illustrated in Figure 4. The more area of the request is acquired, the more profit is generated.

2.2. Formal description

A set of n_r requests originate from the users. These requests lead to handling n_s strips from which $2n_s$ acquisitions are possible. To each request i, are

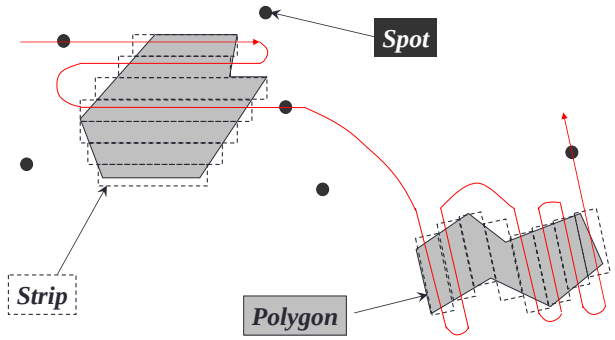

Figure 2: Example of both request's shapes and order for taking the strips after management [19].

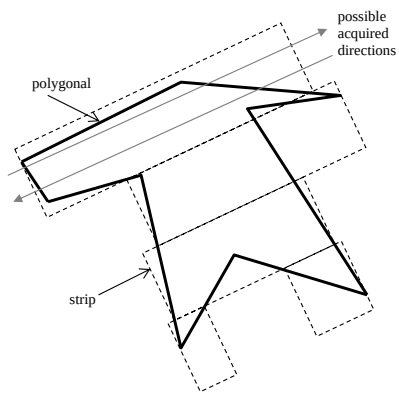

Figure 3: A polygonal area is decomposed into several strips; each strip can be acquired according to two possible directions [19].

associated the user $U[i]_{i=1..n_u}$ who orders it, its gain $G[i]$ for a complete acquisition, its surface area $S[i]$, and its mono/stereo characteristic $St[i] \in \{0 / 1\}$.

The major constraints to fulfill are related to the start time to process a selected acquisition within its time window according to the duration time to acquire it, necessary transition times between two consecutive acquisitions, and satisfying stereo constraints.

A solution is defined by a sequence of selected acquisitions and a sequence of acquisition starting times. Then, given a set of request profits $\{rp_1, rp_2, \ldots, rp_i, \ldots, rp_{n_r}\}$ and a set of user profits $\{up_1, up_2, \ldots, up_j, \ldots, up_{n_u}\}$,

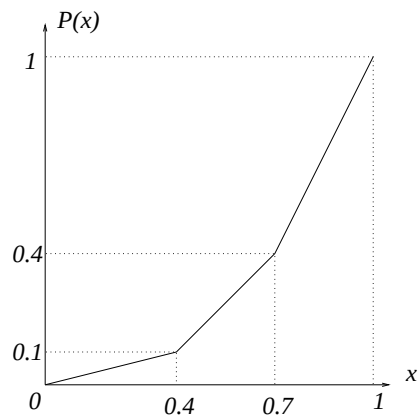

Figure 4: Piecewise linear function of gain $P(x)$ depending on the effective ratio x of acquired area [29].

the following objective functions values are computed:

(1) the total profit:
$\sum_{j=1}^{n_u} up_j$

(2) the maximum profit difference between each pair of users:
$\max_{(j,j')}(|up_j - up_{j'}|)$,
$\forall j \neq j' : up_j = \sum_{i \,|\, U[i]=j} rp_i$,
$\forall i : rp_i = G[i] \cdot S[i] \cdot P(\frac{s_i}{S[i]})$,
where s_i is the area partially taken for request i.

In this work, the set of solutions on the approximate Pareto front will be obtained after solving the model briefly presented above by maximizing the first objective and minimizing the second objective. It should be noted that a lot of objective functions are possible to ensure fairness between the users. Notably, the choice made only compares two users and is not as precise as for instance the minimization to the average. It can also lead to solutions that are of no interest to a decision maker (for instance, the one taking no picture). However, the choice is motivated by the fact that it is easy to

compute, notably in an incremental way. Moreover, the obtained solutions must satisfy the imperative constraints. Then, the decision maker can choose the preferred sequence from the set of solutions on the approximate Pareto front and transmits it to the satellite for operating the mission.

3. Indicator-Based Multi-objective Local Search

In this section, we propose an indicator-based multi-objective local search (IBMOLS) approach to solve the multi-user observation scheduling problem for an agile EOS. IBMOLS is a generic algorithm, which combines the use of basic local search and a binary indicator from IBEA. It was initially proposed in [2].

IBMOLS is an iterated local search. Before starting the first iteration, the approximate Pareto front PO is generated as an empty set and it is updated at the end of each iteration. In each iteration of IBMOLS, it starts by generating the initial population P. Adapted to our problem, two procedures are used in this work. The first one is applied for the first iteration and the second one is applied for the other iterations. Then, the nondominated solutions in the population P are stored in the archive set A. The fitness values of all individuals in the population P are computed by using the indicator based on the hypervolume concept from [33] and the local search step is applied for each individual. After that, the updated population P is combined with the archive set A and the nondominated solutions of this combined set are stored in the new archive set A. If the archive set A changes, the process returns to apply the local search step. Otherwise, this iteration is finished and the final archive set A is obtained. Then, the approximate Pareto front PO will be updated by combining the obtained archive set A with the approximate Pareto front PO, and the set of nondominated solutions from the combined set becomes the new approximate Pareto front PO. If it does not satisfy the stopping criteria, a new initial population P is generated for the next iteration of IBMOLS. The flowchart of the IBMOLS overview is illustrated in Figure 5.

3.1. Population generation – First iteration

For the first iteration of IBMOLS, N individuals are randomly generated to become the initial population. Each individual represents one solution, which is a sequence of selected acquisitions.

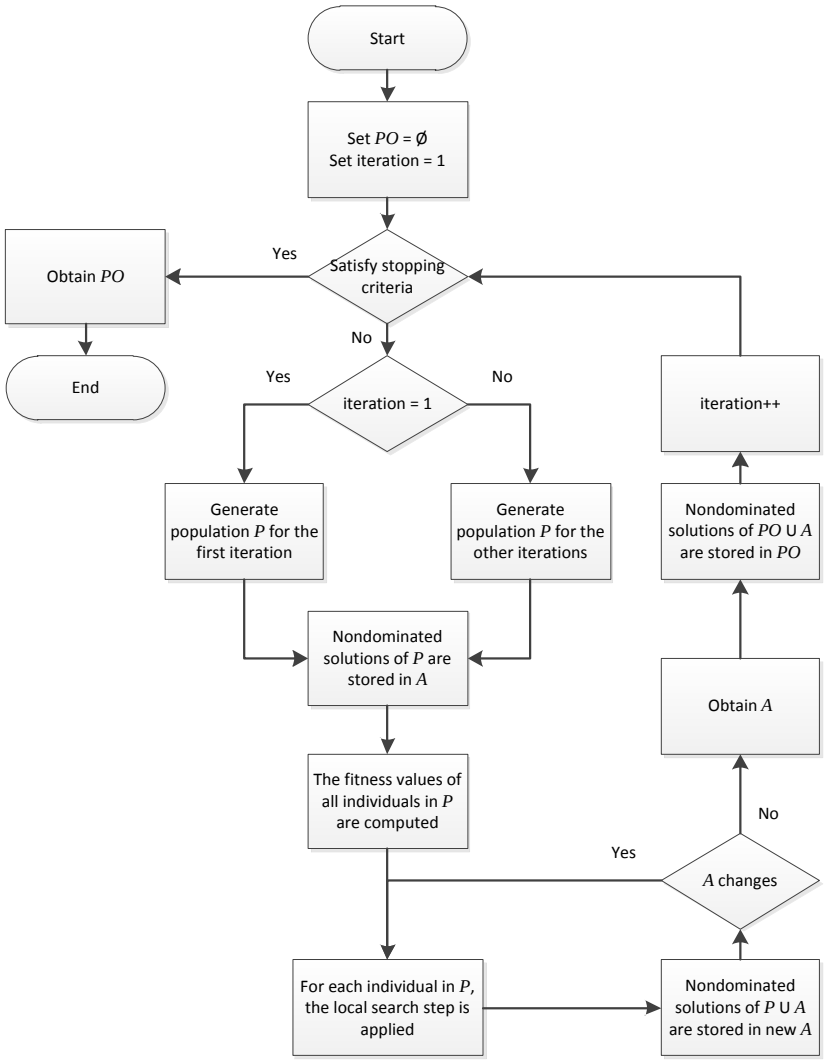

Figure 5: Flowchart of the IBMOLS overview.

The flowchart of the initial population generation for the first iteration is shown in Figure 6. All acquisitions are assigned to be the members of the selected acquisition set. For each acquisition depending on a random order, it is checked whether it satisfies the sufficient transition time constraint and the time window constraint. The solution is coded as a sequence and it is easy to precompute intervals and check feasibility. Note that a more advanced alternative should be to use Simple Temporal Networks [25]. If it satisfies both constraints, the starting time is computed and fixed in the starting time set. Moreover, the acquisition, which concerns the opposite acquired direction of the same strip, is removed from the selected acquisition set. Otherwise, the considered acquisition is removed. The process for checking these constraints is repeated until all acquisitions in the selected acquisition set are tested. After that, the temporary selected acquisition set and starting time set are obtained. In this step, the stereo constraint has to be checked for each selected acquisition one by one in the temporary set. If the considered acquisition comes from the stereo strip, its twin must also be assigned. If its linked acquisition is not assigned, the considered acquisition is removed. When all selected acquisitions are checked, the starting time set is re-computed.

3.2. Population generation – Other iterations

In IBMOLS, an iterated local search is used for searching for the non-dominated solutions by starting the search from different initial populations. A perturbation mechanism is applied in order to escape from local optima. However, the number of modified solution components must be accurately defined. If too high number of modified components is set, better solutions could be found but with a very low probability. On the other hand, if too small number of modified components is defined, the local search will fall back into the local optimum just visited [22].

For the perturbation, an individual is generated by modifying a solution from the approximate Pareto front PO of the current iteration. In this work, the solutions in the approximate Pareto front are randomly selected. The number of selected solutions is equal to the size N of the initial population. Each solution contains the selected acquisition set. It is modified by removing some acquisitions in the random position j from the selected acquisition set. In this work, we set the number of removed elements about 1/4 of the size $\#orig$ of the original selected acquisition set. Moreover, during removing, the stereo constraint has to be checked. If the removed acquisition is a part of

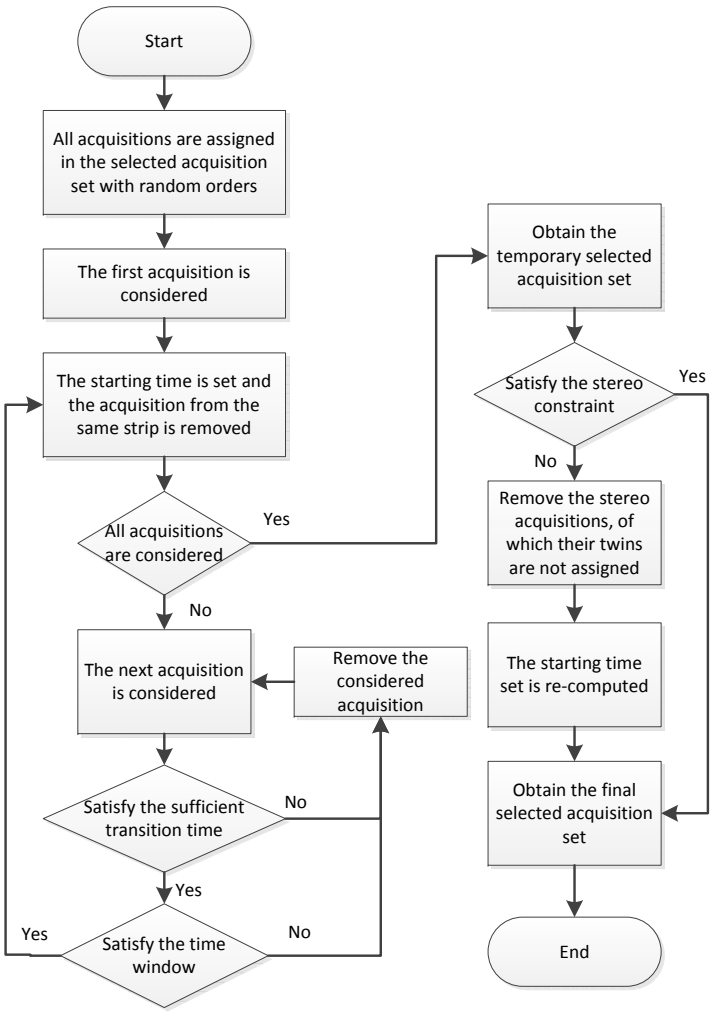

Figure 6: Flowchart of the initial randomly generated population (IBMOLS).

the stereo request, its twin must also be removed. The acquisition removing is repeated until the number $\#modif$ of the remaining acquisitions in the selected acquisition set is less than or equal to 3/4 of the size $\#orig$ of the original set. Then, the modified selected acquisition set will become a part of the individual, which is a member of the initial population in the next iteration. The population generation by using the perturbation is depicted in Algorithm 1.

Algorithm 1 Procedure of the population generating by using the perturbation

for $i := 1$ to N **do**

Step 1: Original individual selection
Select randomly an individual from the approximate Pareto front PO. The selected individual contains the selected acquisition set.

Step 2: Element removing
repeat

Step 2.1: Select randomly the removing position j.
Step 2.2: Remove the acquisition in position j.
Step 2.3: Verify the stereo constraint.
if The removed acquisition is a part of stereo request **then**
Remove the twin of removed acquisition.
end if
until $\#modif \leq \frac{3}{4}\#orig$

Step 3: Counter increment
$i \leftarrow i + 1$
end for

In the process of perturbation, we have to avoid the generation of any already visited solution. Therefore, the number of removed acquisitions is pre-computed. If it is less than or equal to one, the perturbation will generate the individual, which has been visited. In this case, the random generation will be used to generate the individual, instead of using the perturbation.

3.3. Local search step

The local search step starts from an individual in population P and move iteratively to a neighbor. In the original IBMOLS, a first improvement strategy is used to select the neighbor. However, in this work, a best improvement strategy is preferred. During the neighborhood exploration, the fitness values of each neighbor are computed. The neighbor with the best fitness is generated and selected for replacing the worst solution in the population. The neighborhood of all individuals in population P are explored. The population and the archive set are updated. If the updated archive set A does not change, the local search step will be stopped. Otherwise, another local search step is performed.

From the selected solution in the archive, the best neighbor is searched as follows. Each acquisition is considered in a given order. If the acquisition is scheduled in the sequence, the process is straightforward: removing the acquisition just leads to the new solution objectives to be updated. If the acquisition is a stereo one, both scheduled acquisitions are removed from the sequence. If the acquisition is not in the sequence, we try to insert it in the sequence. In the case of a stereo acquisitions, both tasks are tested for inclusion but the principle remains the same as follows.

The acquisition is tried for insertion between two scheduled ones in the sequence. However, the feasibility of the insertion has to be checked as the operational constraints must hold. The insertion of an acquisition implies that the already scheduled ones after the insertion place may need to be shifted to the right, delaying them as illustrated in Figure 7. This can be checked easily by a precomputing process. For a scheduled acquisition id, the latest possible starting time $Q(id)$ is computed as follows. For the last scheduled acquisition sa_N, its latest starting time $Q(N)$ can be computed by

$$Q(N) = T_{max}(sa_N)$$

where T_{max} denotes the latest starting time of an acquisition.

For the other acquisitions sa_{id}, where $1 \leq id \leq N-1$, their latest starting times $Q(id)$ are given by

$$Q(id) = \min(T_{max}(sa_{id}), Q(id+1) - Du(sa_{id+1}) - Dt(sa_{id}, sa_{id+1}))$$

where Du and Dt stand for the duration time of an acquisition and the transition time between two acquisitions, respectively.

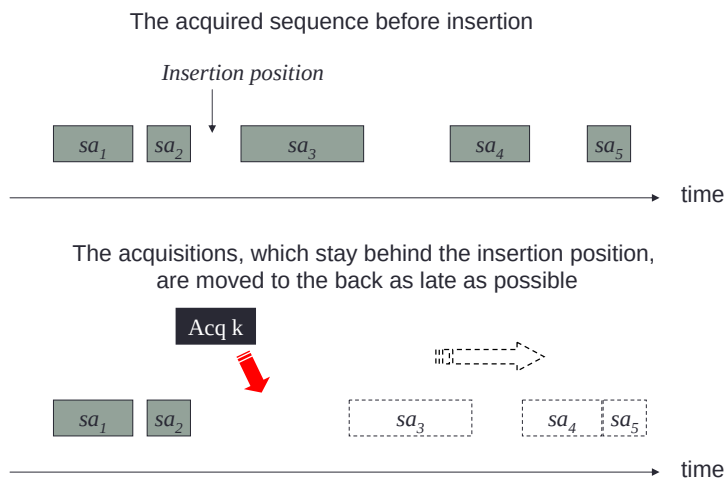

Figure 7: A large space is prepared for the insertion of the unassigned acquisition.

Example of the latest starting time $Q(id)$ computation is shown in Figure 8.

Then the possibility to insert an acquisition $Acq\ k$ is tested using the following rules:

- **the acquisition $Acq\ k$ can be inserted in the first position (position 1) of the selected acquisition set, only if**

$$T_{min}(Acq\ k) + Du(Acq\ k) + Dt(Acq\ k, sa_1) \leq Q(1)$$

- **the acquisition $Acq\ k$ can be inserted in the middle position id (position 2 to N) of the selected acquisition set, only if**

$$ta_{id-1} + Du(sa_{id-1}) + Dt(sa_{id-1}, Acq\ k) \leq T_{max}(Acq\ k)$$

  and

$$Tk + Du(Acq\ k) + Dt(Acq\ k, sa_{id}) \leq Q(id)$$

  where $Tk = \max(T_{min}(Acq\ k), ta_{id-1} + Du(sa_{id-1}) + Dt(sa_{id-1}, Acq\ k))$

- **the acquisition $Acq\ k$ can be inserted in the last position (position $N+1$) of the selected acquisition set, only if**

$$ta_N + Du(sa_N) + Dt(sa_N, Acq\ k) \leq T_{max}(Acq\ k)$$

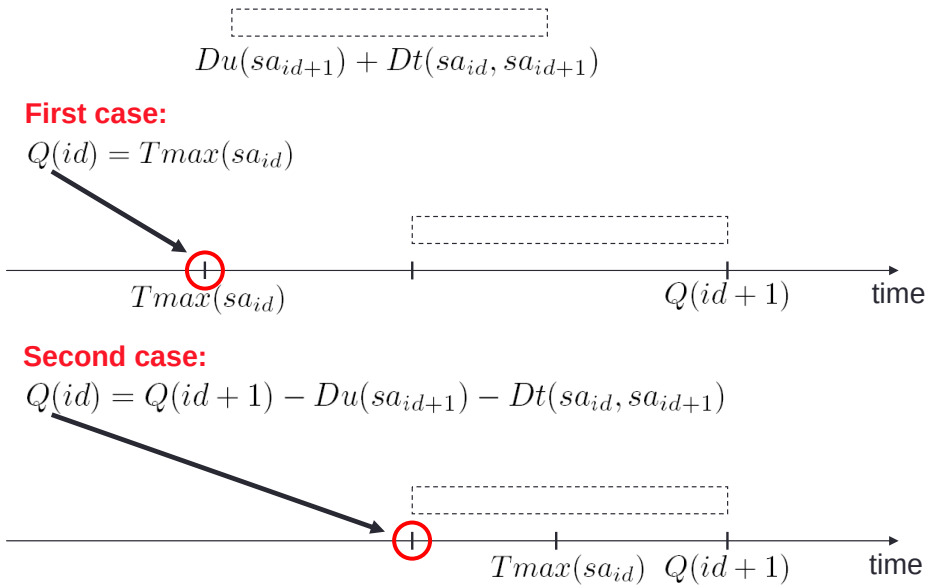

Figure 8: Example of the latest starting time $Q(id)$ computation

4. Computational Results

The methods were tested on modified instances from the ROADEF 2003 challenge (`http://challenge.roadef.org/2003/en/sujet.php`) (TestSet A, B, and X). They are modified for 4-user requirements and the format of instance names are changed to `a_b_c`, where `a` is the number of requests, `b` is the number of stereo requests, and `c` is the number of strips. The modified instances are publicly available and can be downloaded from: `http://www.laas.fr/files/ROC/4U-EOS.zip`.

4.1. Biased random-key genetic algorithm

We presented the biased random-key genetic algorithm (BRKGA) for solving this Earth observation scheduling problem in [27][28]. Genetic algorithms are metaheuristic search methods, which can solve large-size problem instances and obtain satisfying solutions in an acceptable time [26]. They start by generating a population of p chromosomes, and then involve three mechanisms: selection, crossover, and mutation, to generate the new chromosomes for the next generation and repeat the iterations until some stopping conditions are satisfied. BRKGA combines genetic algorithm and the

concept of random key [11]. It has different ways to select two parents for the crossover operation, compared with the original of random-key genetic algorithm (RKGA) [4]. In [27][28], parameter values of BRKGA were experimentally tuned. The population size of BRKGA was set equal to the length of the random-key chromosome or twice the number of strips. For next population generation of BRKGA, the size of the three parts (the elite set, the mutant set, and the crossover offspring set) was set in accordance with the recommended values in [12]. For the elite set part, all the objective functions must be taken into account for the fitness of each chromosome. Thus, we chose a selecting method from several efficient algorithms, e.g., Nondominated Sorting Genetic Algorithm (NSGA-II) [9], $\mathscr{S}$ *metric selection* Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [5], Indicator-Based Evolutionary Algorithm (IBEA) [33] in order to use in the selection process. The size of the elite set was equal to the number of non-repeated schedules from the nondominated solutions, but it was not over $0.15p$. The size of mutant set was equal to $0.3p$. The probability of elite element inheritance for crossover operation was set to 0.6. Moreover, BRKGA used encoding and decoding steps for managing the solution. For the decoding step, the basic decoding was implemented in [27]. Several decoding procedures have been proposed in [27, 28]. We will test IBMOLS agains the best version of the algorithm.

In each iteration of BRKGA, the nondominated solutions were stored in an archive. If there was at least one solution from the current population that can dominate some solutions in the archive, the archive was updated. Therefore, we used the number of iterations since the latest archive improvement to be a stopping criterion. The stopping value was set to 50. The hybrid decoding method obtained the best solutions.

4.2. IBMOLS evaluation

In this work, the indicator-based multi-objective local search (IBMOLS) is used to solve the Earth observation scheduling problem. The obtained results are compared with the results from BRKGA. Some parameter values have to be tuned. Thus, we set a value of 10 for the population sizes. Moreover, as previously, we chose a number of 50 iterations since the latest approximate Pareto set improvement to be the stopping criterion.

As suggested in [14], the hyper volume metric [32] was used. The hypervolume represents the size of the area dominated by the

approximation generated by an algorithm. It is based on computing the volume (area in the bi-objective case) dominated by a given Pareto-front approximation. The hypervolume metric requires a reference point consisting of a reference value for each of the two objectives. Here we use 0 for total profit maximization and the maximum sum of profits per user for the second objective. These values are the worst possible values for each objective.

Both proposed algorithms are implemented in C++ and thirty runs per instance are tested. Hypervolumes of the approximate Pareto front are computed by using a reference point of 0 for the first objective (maximizing the total profit) and the maximum of the profit summations of each user for the second one (minimizing the profit difference between users). The hypervolume values, which are obtained from both proposed algorithms, are plotted by box plots.

We compare the results between BRKGA and IBMOLS. For BRKGA, the results were obtained from the hybrid decoding, which used the selection method of indicator-based evolutionary algorithm (IBEA) for selecting the preferred chromosomes to become the elite set. Moreover, the selection of the dominant solution was used to manage the elite set in the hybrid decoding process. For IBMOLS, the results are obtained by using the neighborhood structure, which consists of the insertion and removing of the mono and stereo acquisitions. The method of feasibility checking, which computes the latest starting time before checking insertion feasibility, is applied. For generating the initial population, the random generation is used in the first iteration and the perturbation is applied in the other iterations. The number of 50 iterations since the latest archive improvement is used as a stopping criterion for both BRKGA and IBMOLS. The box plots of hypervolume values and the average computation time of BRKGA and IBMOLS for TestSet A, B, and X are presented in Figure 9, 10, and 11, respectively. For each instance, the first column illustrates the results from BRKGA and the second column shows the results from IBMOLS. Moreover, we also use a Mann-Whitney statistical test [15] for comparing the results from both algorithms.

For BRKGA, the results of the smallest instance (instance `2_0_2`) cannot be reached. Indeed, the population size, which is equal to twice of the number of strips, is too small for generating the new generation from the three sets of chromosomes in BRKGA process.

In the comparison, the box plots show that IBMOLS obtains better median values of the hypervolume for all instances and better standard devia-

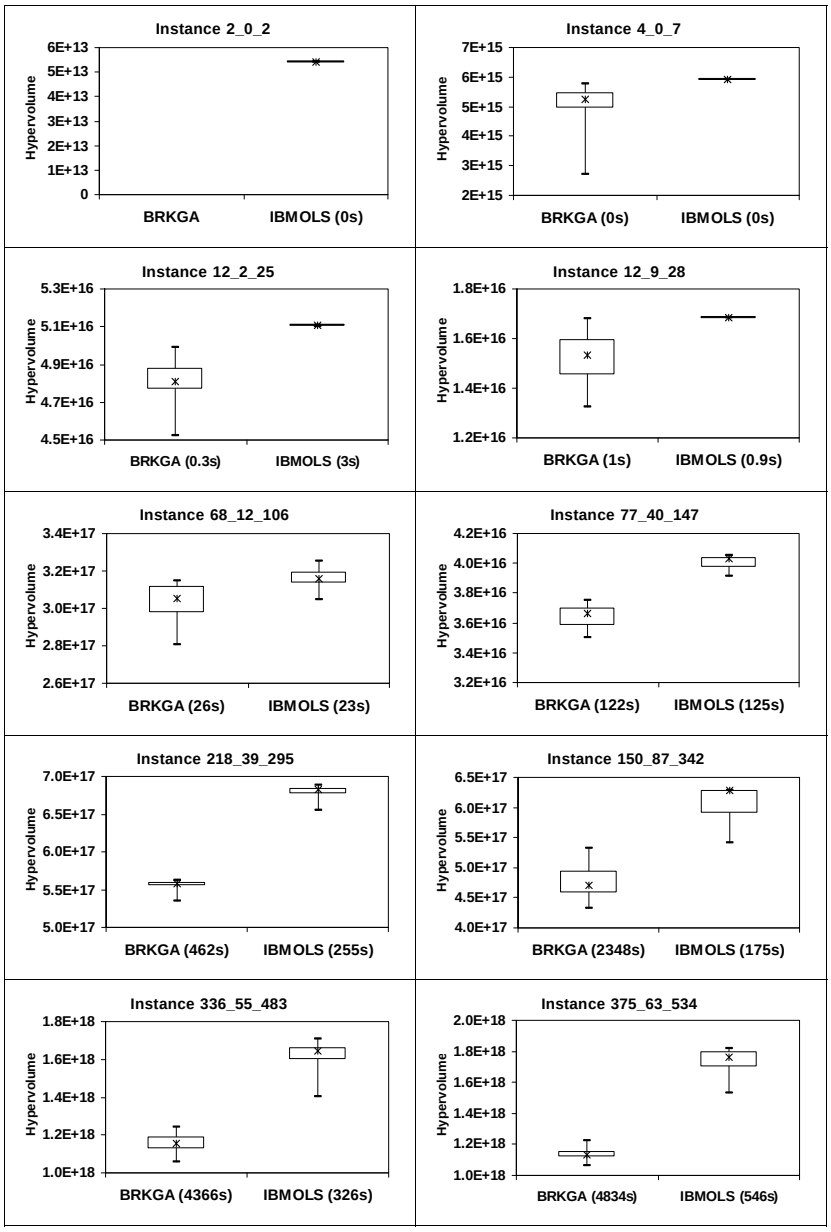

Figure 9: Comparison of hypervolume values of the approximate Pareto front between BRKGA and IBMOLS of TestSet A (for each method, the average computation times are indicated in parentheses).

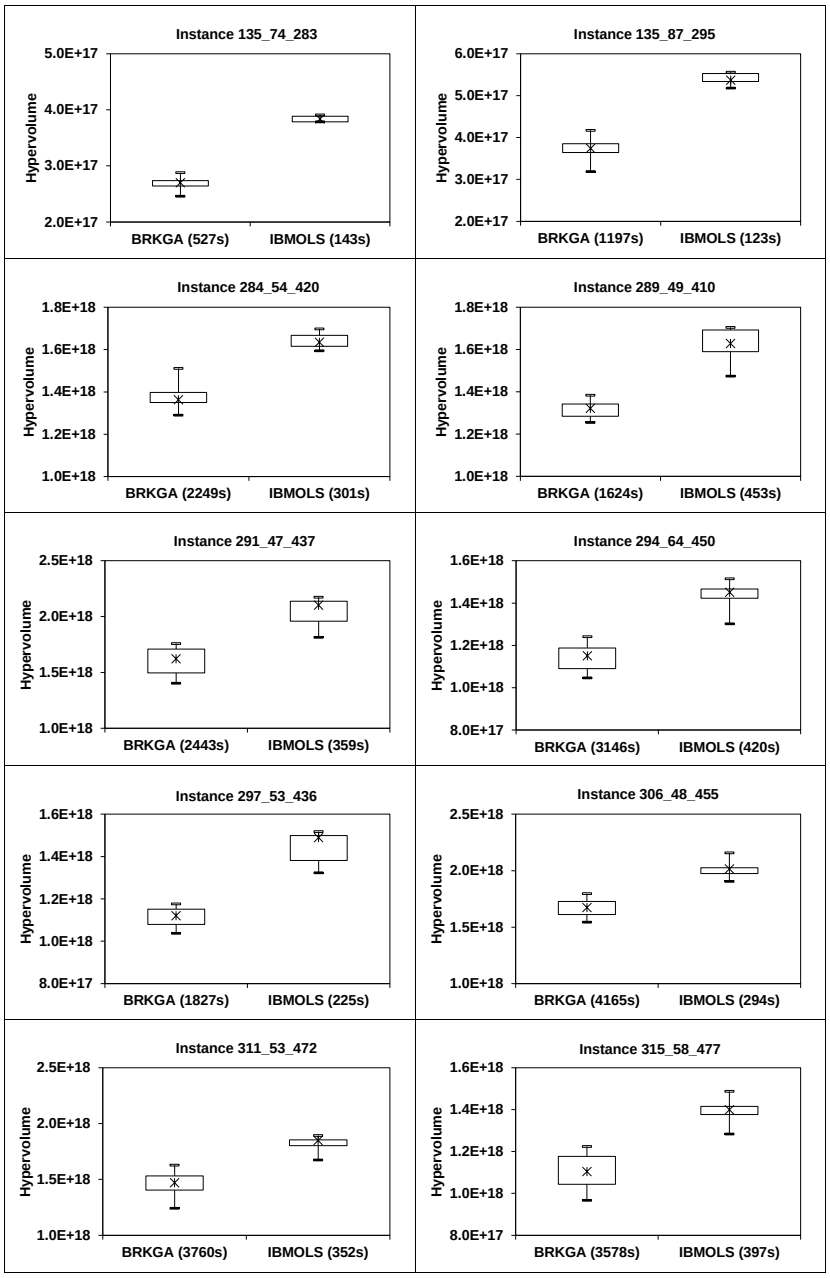

Figure 10: Comparison of hypervolume values of the approximate Pareto front between BRKGA and IBMOLS of TestSet B (for each method, the average computation times are indicated in parentheses).

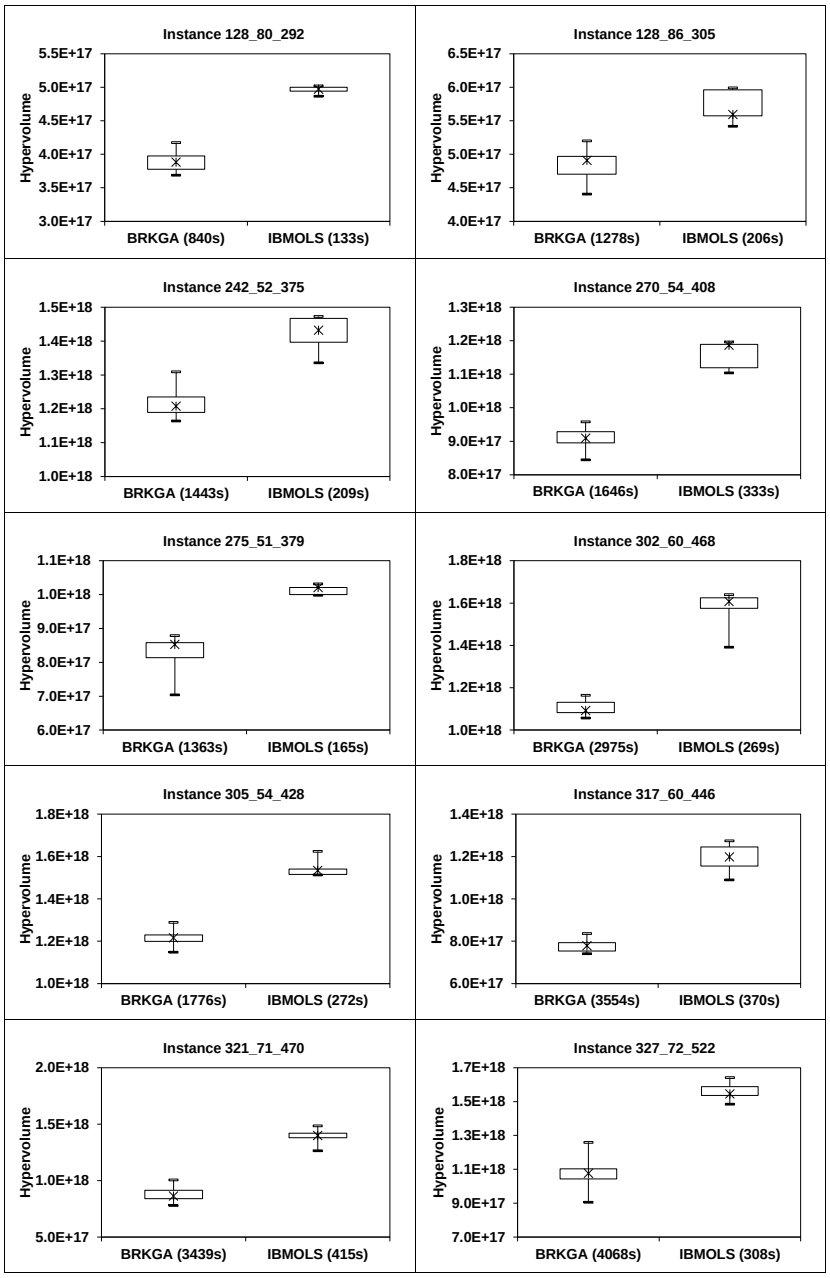

Figure 11: Comparison of hypervolume values of the approximate Pareto front between BRKGA and IBMOLS of TestSet X (for each method, the average computation times are indicated in parentheses).

tions for most of the results. Moreover, IBMOLS spends less computation time than BRKGA, especially for large instances. Additionally, the statistical test reveals that the results of IBMOLS are significantly better than those of BRKGA. Figure 12 illustrates the improvement of the hypervolume values versus the computation times for some instances of TestSet A, B, and X. In each graph, the improvement of hypervolume values between BRKGA process and IBMOLS process is analyzed. The results show that IBMOLS obtains solutions closer to the Pareto optimal solutions for medium and large instances. Moreover, it can also converge to the Pareto optimal solutions faster than BRKGA.

Finally, the best approximate Pareto fronts of some instances in TestSet A, B and X, which are obtained from IBMOLS, are illustrated in Figure 13. For each instance, the total profit is presented on x-axis and the maximum profit difference between users is presented on y-axis. The hypervolume value of each approximate Pareto front is also indicated. The figure shows that IBMOLS can obtain good nondominated solutions on the approximate Pareto fronts for all instances. Moreover, the best total profit values of TestSet A, B, and X, which are obtained from IBMOLS and from the winners of ROADEF 2003 challenge [8][16], are compared in Table 1. Although our work considers two objective functions simultaneously, IBMOLS can reach values of best total profit, the values of which are quite close to the results from the winners of ROADEF 2003 challenge. Indeed, for TestSets A, B, and X, the relative deviation of IBMOLS is 1.7%, 3.7%, and 3.4%, respectively; over all instances, this deviation is then only 3.0%.

5. Conclusions and future works

An indicator-based multi-objective local search (IBMOLS) is used for solving a multi-objective optimization problem associated with selecting and scheduling observations of an agile Earth observing satellite. The ROADEF 2003 challenge instances are modified in order to take account explicitly of 4-user requirements. Two objective functions, maximizing the total profit and minimizing the maximum profit difference between users for the fairness of resource sharing, are considered and the imperative constraints must be satisfied. The implementation was done and the results, which are obtained from IBMOLS and the biased random-key genetic algorithm (BRKGA), are compared.

For BRKGA, random-key encoding generates each chromosome in the

population and the chromosomes are decoded to be the sequences of the selected acquisitions. A hybrid decoding is presented in this paper. The elite selecting method from IBEA is used for selecting the preferred solutions to become the elite set of the population. An elite set, a crossover offspring set, and a mutant set are combined to become the next population. The selection of the dominant solution is used to manage the elite set in the hybrid decoding process.

IBMOLS is implemented by generating the initial population using a random generation for the first iteration and a perturbation for the other iterations. The indicator-based fitness assignment with the hypervolume concept from IBEA is applied for comparing the solutions in the population. The hypervolume values from IBMOLS and BRKGA are compared. Most of the results show that IBMOLS obtains better solutions and it spends less computation time.

As perspectives, we present short term and long term further works. Concerning the short term works, it may be possible to improve the results obtained from IBMOLS. The initial population generation step for the first iteration by using data of the problem instances may be used instead of the random generation. For example, the order to assign each acquisition in the initial sequence can depend on the acquiring duration time of each acquisition. The acquisition with the longest acquiring duration time should be assigned firstly in the sequence. Moreover, the number of removed elements in the perturbation for the initial population generation in the other iterations can also be modified. For the long term works, other perturbation rules and other neighborhood structures can be explored. An example of perturbation rule is to insert some feasible acquisitions for replacing the removed elements.

Acknowledgments

This research was partially funded by the THEOS Operational Training Programme (TOTP) of Geo-Informatics and Space Technology Development Agency in Thailand (GISTDA). This research was initiated while the first author was with LAAS-CNRS, Université de Toulouse, France. The authors are indebted to anonymous referees for their numerous constructive remarks.

References

[1] Arias-Montaño, A., Coello, C.A.C., & Mezura-Montes, E. (2012). Multi-objective Evolutionary Algorithms in Aeronautical and Aerospace Engineering. *IEEE Transactions on Evolutionary Computation*, 16, 662–694.

[2] Basseur, M., & Burke, E.K. (2007). Indicator-based multi-objective local search. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, (pp. 3100-3107).

[3] Bataille, N., Lemaître, M., & Verfaillie, G. (1999). Efficiency and Fairness when Sharing the Use of a Satellite. In *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, The Netherlands, (pp. 465–470).

[4] Bean, J.C. (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization. *ORSA Journal on Computing*, 6, 154–160.

[5] Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multi-objective selection based on dominated hypervolume. *European Journal of Operational Research*, 181, 1653–1669.

[6] Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G., & Raymond, V. (2007). A Heuristic for the Multi-Satellite, Multi-Orbit and Multi-User Management of Earth Observation Satellites. *European Journal of Operational Research*, 177, 750–762.

[7] Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified Tabu Search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52, 928–936.

[8] Cordeau, J.-F., & Laporte, G. (2005). Maximizing the Value of an Earth Observation Satellite Orbit. *Journal of the Operational Research Society*, 56, 962–968.

[9] Deb, K., Pratep, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.

[10] 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, 533–542.

[11] Gonçalves, J.-F., & de Almeida, J. (2002). A Hybrid Genetic Algorithm for Assembly Line Balancing. *Journal of Heuristics*, 8, 629–642.

[12] Gonçalves, J.-F., & Resende, M.G.C. (2011). Biased Random-Key Genetic Algorithms for Combinatorial Optimization. *Journal of Heuristics*, 17, 487–525.

[13] Habet, D., Vasquez, M., & Vimont, Y. (2010). Bounding the optimum for the problem of scheduling the photographs of an Agile Earth Observing Satellite. *Computational Optimization and Applications*, 47, 307–333.

[14] Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, (pp. 711–726).

[15] Knowles, J., Thiele, L., & Zitzler, E. (2006). *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.

[16] Kuipers, E.J. (2003). An Algorithm for Selecting and Timetabling Requests for an Earth Observation Satellite. *Bulletin de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, 7–10.

[17] Lemaître, M., Verfaillie, G., & Bataille, N. (1999). Exploiting a Common Property Resource under a Fairness Constraint: a Case Study. In *Proceedings of the 16th international joint conference on Artifical intelligence (IJCAI'99)*, Stockholm, Sweden, vol. 1 (pp. 206–211).

[18] Lemaître, M., Verfaillie, G., & Jouhaud, F. (2000). How to manage the new generation of Agile Earth Observation Satellites. In *6th International SpaceOps Symposium (Space Operations)*, Toulouse, France.

[19] Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., & Bataille, N. (2002). Selecting and Scheduling Observations of Agile Satellites. *Aerospace Science and Technology*, 6, 367–381.

[20] Lemaître, M., Verfaillie, G., Fargier, H., Lang, J., Bataille, N., & Lachiver, J.-M. (2002). Sharing the use of Earth observation satellites. *3rd NASA Workshop on Planning and Scheduling*, Houston, Texas, USA.

[21] Li, Y., Xu, M., & Wang, R. (2007). Scheduling Observations of Agile Satellites with Combined Genetic Algorithm. In *Proceedings of the 3rd International Conference on Natural Computation (ICNC 2007)*, Haikou, China, vol. 3 (pp. 29–33).

[22] Lourenço, H.R., Martin, O.C., & Stützle, T. (2002). Iterated local search. In Glover, F., Kochenberger, G. (Eds.), *Handbook of Metaheuristics* (pp. 321–353). Kluwer Academic Publishers, Norwell, MA.

[23] Mansour, M.A.A., & Dessouky, M.M. (2010). A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite. *Computers & Industrial Engineering*, 58, 509–520.

[24] Mendes, J.J.M., Gonçalves, J.-F., & Resende, M.G.C. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers and Operations Research*, 36, 92–109.

[25] Pralet, C., & Verfaillie, G. (2013). Time-dependent simple temporal networks: properties and algorithms. *RAIRO Operations Research*, 47, 173–198.

[26] Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation.* Wiley.

[27] Tangpattanakul, P., Jozefowiez, N., & Lopez, P. (2012). Multi-objective optimization for selecting and scheduling observations by agile Earth observing satellites. In Coello Coello, C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (Eds.), *PPSN XII. LNCS*, vol. 7492 (pp. 112–121). Springer, Heidelberg.

[28] Tangpattanakul, P., Jozefowiez, N., & Lopez, P. (2013). Biased random key genetic algorithm with hybrid decoding for multi-objective optimization. In *Proceedings of Federated Conference on Computer Sciences and Information Systems 2013*, Krakow, Poland, September 2013.

[29] Verfaillie, G., Lemaître, M., Bataille, N., & Lachiver, J.-M. (2002). *Management of the Mission of Earth Observation Satellites Challenge Description*. Technical report, Centre National d'Etudes Spatiales, France.

[30] Wang, J., Jing, N., Li, J., & Chen, Z.H. (2007). A multi-objective imaging scheduling approach for earth observing satellites. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07)*, London, England, (pp. 2211–2218).

[31] Yager, R.R. (1988). On Order Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems Man and Cybernetics*, 18, 183–190.

[32] Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications. *Ph.D. thesis*, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

[33] Zitzler, E., & Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., & Schwefel, H.-P. (Eds.), *PPSN VIII. LNCS*, vol. 3242 (pp. 832–842). Springer, Heidelberg.

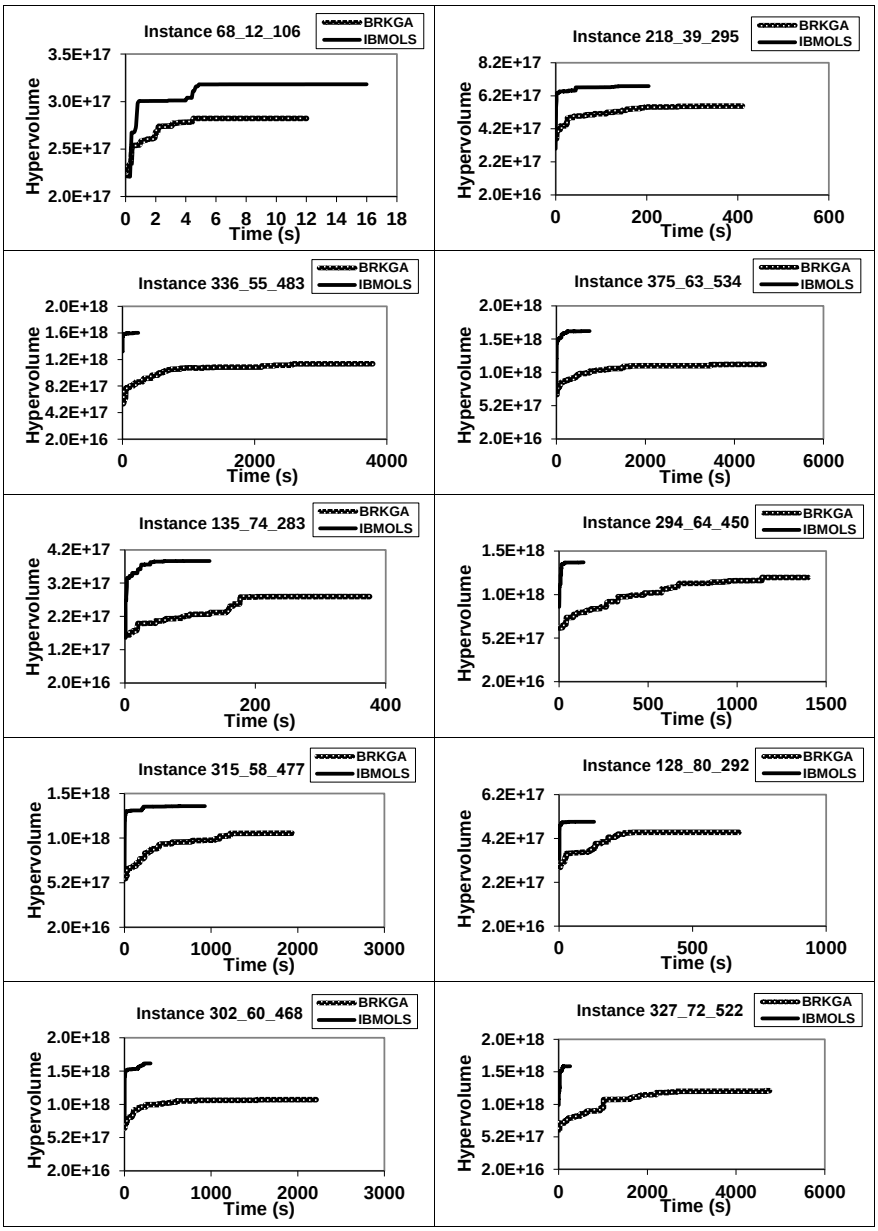

Figure 12: Comparison of the improvement of hypervolume values versus the computation times between BRKGA process and IBMOLS process from some TestSet A, B, and X instances when using the fixed iterations since the latest archive improvement as the stopping criterion.

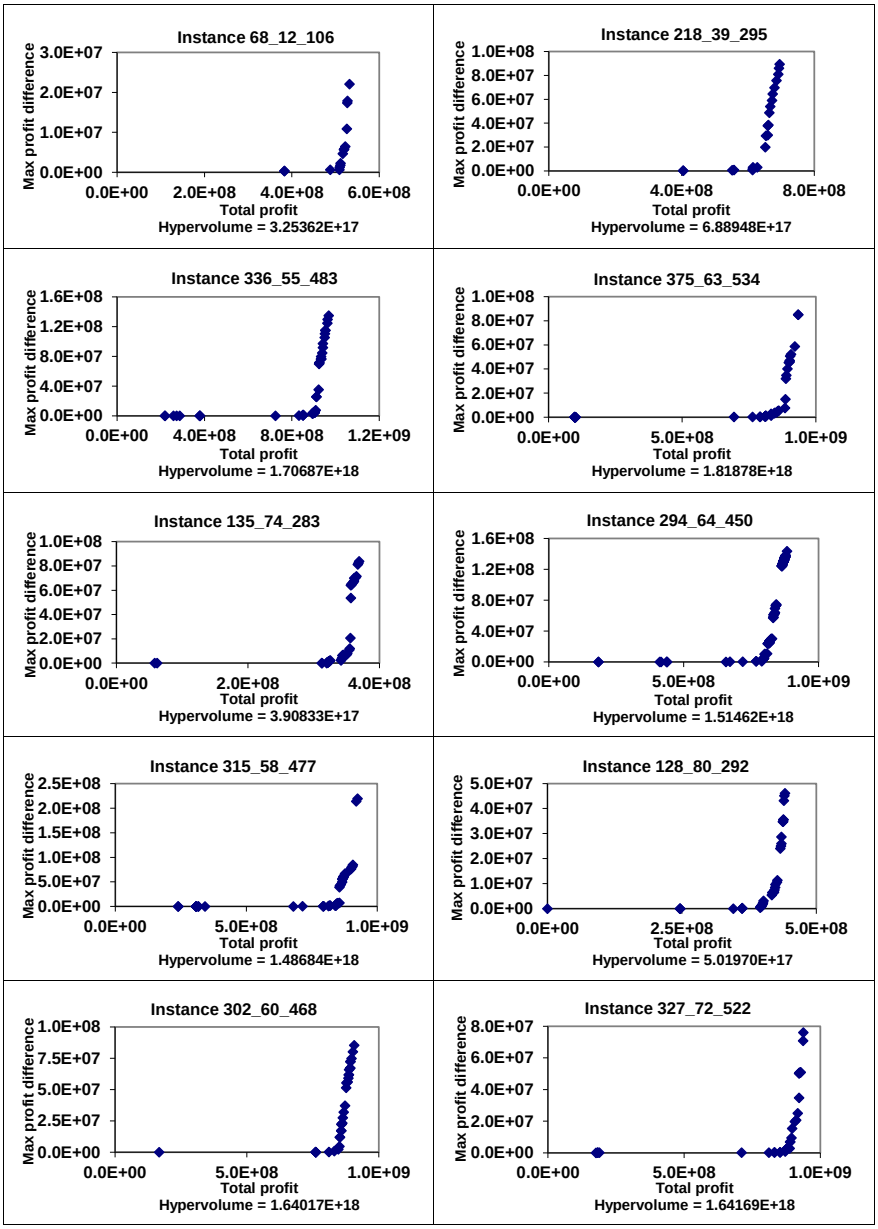

Figure 13: The best approximate Pareto front of some TestSet A, B, and X instances.

Instance		Best total profit value	
		ROADEF 03	IBMOLS
TestSet A	2_0_2	1.04234×10^7	1.04234×10^7
	4_0_7	1.15711×10^8	1.15711×10^8
	12_2_25	1.91358×10^8	1.91358×10^8
	12_9_28	1.21680×10^8	1.21680×10^8
	68_12_106	5.63597×10^8	5.27115×10^8
	77_40_147	1.85407×10^8	1.84904×10^8
	218_39_295	7.17384×10^8	6.93287×10^8
	150_87_342	4.25983×10^8	4.25983×10^8
	336_55_483	1.00530×10^9	9.67989×10^8
	375_63_534	9.67911×10^8	9.34503×10^8
TestSet B	135_74_283	3.73551×10^8	3.68789×10^8
	135_87_295	4.60197×10^8	4.47650×10^8
	284_54_420	1.03006×10^9	9.75143×10^8
	289_49_410	8.42379×10^8	8.02015×10^8
	291_47_437	1.14332×10^9	1.08436×10^9
	294_64_450	9.15035×10^8	8.83632×10^8
	297_53_436	9.86678×10^8	9.57178×10^8
	306_48_455	1.16178×10^9	1.11783×10^9
	311_53_472	1.15063×10^9	1.09868×10^9
	315_58_477	9.54966×10^8	9.24819×10^8
TestSet X	128_80_292	4.58107×10^8	4.41712×10^8
	128_86_305	4.62070×10^8	4.50544×10^8
	242_52_375	9.45737×10^8	9.00455×10^8
	270_54_408	8.75447×10^8	8.36552×10^8
	275_51_379	8.33287×10^8	8.22863×10^8
	302_60_468	9.63809×10^8	9.06681×10^8
	305_54_428	9.77811×10^8	9.65784×10^8
	317_60_446	8.78848×10^8	8.73636×10^8
	321_71_470	9.52267×10^8	9.07546×10^8
	327_72_522	9.92155×10^8	9.37968×10^8

Table 1: Comparison of the best total profit value of TestSet A, B, and X, which are obtained from IBMOLS and from the winners of ROADEF 2003 challenge.