Multi-objective Optimization for Selecting and Scheduling Observations by Agile Earth Observing Satellites

Panwadee Tangpattanakul[1,2], Nicolas Jozefowiez[1,3], and Pierre Lopez[1,2]

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2] Univ de Toulouse, LAAS, F-31400 Toulouse, France
[3] Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
{panwadee.tangpattanakul,nicolas.jozefowiez,pierre.lopez}@laas.fr

Abstract. This paper presents a biased random-key genetic algorithm for solving a multi-objective optimization problem concerning the management of agile Earth observing satellites. It addresses the selection and scheduling of a subset of photographs from a set of candidates in order to optimize two objectives: maximizing the total profit, and ensuring fairness among users by minimizing the maximum profit difference between users. Two methods, one based on dominance, the other based on indicator, are compared to select the preferred solutions. The methods are evaluated on realistic instances derived from the 2003 ROADEF challenge.

Keywords: Multi-objective optimization, Earth observing satellite, scheduling, genetic algorithm.

1 Introduction

This paper studies the use of multiobjective optimization applied to the scheduling of one Earth observing satellite in a context where multiple users request photographs from the satellite. Genetic algorithms are 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 satisfying users' requirements. When the ground station center receives requests from several users, it has to consider all users' requirements and output an order consisting of a sequence of selected photographs to be transmitted to the satellites. The management problem of EOSs is to select and schedule a subset of photographs from a set of candidates. Among the various types of EOSs, only agile satellites are considered in our study.

An agile EOS has one on-board camera that can move in three axes: roll, pitch, and yaw. It has more efficient capabilities for taking photographs than for example, SPOT5, a non-agile satellite. The selection and scheduling of taking photographs with agile EOSs is more complicated because there are several possible schedules for the same set of selected photographs. The starting time of

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 112–121, 2012.

each photograph is not fixed; nonetheless, it must be within a given time interval. This problem is a scheduling problem and it is the issue under consideration in this paper.

Several algorithms including greedy algorithm, dynamic programming, constraint programming, and local search have been applied for solving agile EOSs scheduling problems [1]. The ROADEF 2003 challenge (see http://challenge.roadef.org/2003/en/) requires the scheduling solutions that maximize total profit of the acquired photographs and also satisfy all physical constraints of agile EOSs. The winner used an algorithm based on simulated annealing [2] and the second prize winner proposed an algorithm based on tabu search [3].

Our work considers agile EOSs scheduling problem where the requests emanate from different users. Hence an objective function to maximize the total profit is not sufficient. The ground station center should also share fairly the resources among users. Therefore, a multi-objective model is considered. The idea to use two objective functions related to fairness and efficiency was proposed in [4], and 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 between fairness and efficiency. For the multi-criteria method, instead of building a complete set of non-dominated solutions, the authors only searched for a decision close to the line with a specified slope on the objective function plane. In [5], a tabu search was used for the multi-satellite, multi-orbit, and multi-user management to select and schedule requests. The upper bounds on the profit were derived by means of a column generation technique. They tested these algorithms with the data instances provided by the French Center for Spatial Studies (CNES).

This paper proposes a biased random-key genetic algorithm (BRKGA) in order to solve the multi-objective optimization problem for selecting and scheduling the subset of required photographs from multiple users. The two objective functions for this scheduling problem are to maximize the total profit and minimize the maximum difference of profit values between users. The second objective function represents the fairness of resources sharing among the users. The solutions must also satisfy the physical constraints of the agile EOSs.

The article is organized as follows. The problem is explained in Section 2. Section 3 describes the biased random-key genetic algorithm for solving the multi-objective optimization problem. The computational results are reported in Section 4. Finally, conclusions and future work are discussed in Section 5.

2 Multi-objective Optimization for Photograph Scheduling Problem of Agile Earth Observing Satellites

According to the mission and physical constraints of agile EOSs, the requests which are required from users cannot be assigned to a satellite directly. The shape of the area of candidate photographs can be either a spot or polygonal. A spot is a small circular area with a radius of less than 10 km. A polygonal area is an area ranging from 20 to 100 km. All requests (both spot and polygonal

area) must be managed from the ground station center by transforming the requests into rectangular shapes called strips for which the camera can take a photograph at once. Each spot is considered as one single strip. Each polygonal area is decomposed into several strips with fixed width but variable length. Each strip can be acquired following two possible opposite directions as shown in Figure 1, only one of them will be selected in the scheduling results. Requests can be mono or stereo photographs. A mono photograph is taken only once, whereas a stereo photograph must be acquired twice in the same direction but from different angles.

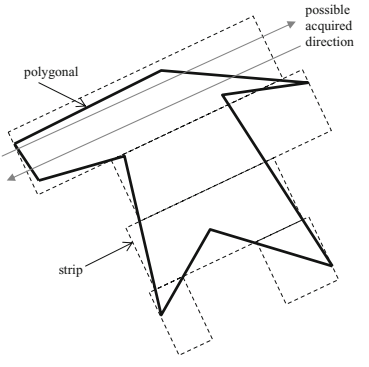

Fig. 1. A polygonal area is decomposed into several strips; each strip can be acquired according to two possible directions

In [6], a simplified version of the problem of managing the mission of agile Earth observing satellites was presented. An instance gives the set of candidate requests with shape type, mono or stereo characteristic, associated gain and surface areas. Let r be the set of requests. These requests are divided into the set of strips s. Each strip includes details, which consist of the identity of request $R[j]$ where that strip is split from, the useful surface area $Su[j]$, duration time $Du[j]$, and earliest and latest visible times from two ends $Te[j,0], Tl[j,0], Te[j,1]$, and $Tl[j,1]$. Each strip is possibly taken from two directions but only one can be selected. Thus, our scheduling problem is solved for selecting and scheduling the possible strip acquisition that is associated with the possible acquisition direction of each strip. If one possible strip acquisition is selected, the other one (possible acquisition in opposite direction) of the same strip is forbidden to be selected. For the profit calculation of each acquired request, its profit can be calculated by a piecewise linear function of gain depending on the fraction of taken useful area and the whole area of each request, as illustrated in Figure 2.

Hence, we extend the case to multiple users as in [5]. However, we solve the problem as a real bi-objective problem. The two objectives are to maximize total profit and ensure fairness between users. For the second objective, the defined function is to minimize the maximum difference in profit between the users. The imperative constraints for finding the feasible solutions are: take each strip within

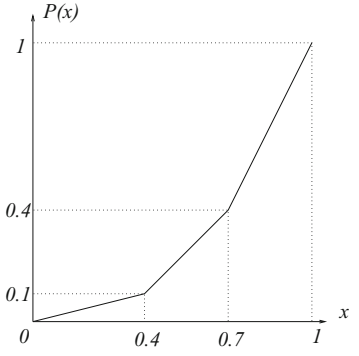

Fig. 2. Piecewise linear function of gain $P(x)$ depending on the effective ratio x of acquired area [6]

their associated time windows, no overlapping images, sufficient transition times, acquire only one direction for each strip, and satisfy the stereoscopic constraint for stereo requests.

3 Biased Random-key Genetic Algorithm for the Multi-user Photograph Scheduling

Genetic algorithm is a heuristic search method that mimics the process of natural evolution. The starting step of genetic algorithm is the initial population generation and the population consists of several chromosomes. Each chromosome, which is formed of several genes, represents one solution. The genetic algorithm involves three mechanisms (selection, crossover, and mutation) to generate the new chromosomes for the next generation and repeats to generate the new generation until the stopping criterion is satisfied.

We propose a genetic algorithm for selecting and scheduling the required photographs for the agile EOSs from multi-user requests. The biased random-key genetic algorithm (BRKGA) [7] is used to solve this scheduling problem with two important steps (encoding and decoding). Two methods are used to select the preferred solutions in each genetic algorithm iteration: i) fast nondominated sorting with crowding distance assignment [8]; ii) indicator based on the hypervolume concept [9]. Let p, p_e, and p_m be the sizes of the population, of the elite set, and of the mutation set, respectively.

3.1 Chromosome Generation in the Encoding Process

The initial population consisting of p chromosomes is generated. Each chromosome consists of genes which are encoded by real values randomly generated in the interval $(0, 1]$. For our problem, each strip can be taken following two opposite directions (but only one direction will be selected). Each gene is associated

with one direction of a strip, which is a possible strip acquisition that we will just call *acquisition* in the sequel, for the sake of simplification. It is the reason why we define the number of genes to be equal to twice the number of strips.

3.2 Schedule Generation in the Decoding Process

Each chromosome is decoded in order to obtain one solution, which is a sequence of selected acquisitions of the scheduling problem. In this decoding step, the considered priority of each acquisition depends on the gene values: from high to low. The imperative constraints, except the stereo constraint, are verified during each considered acquisition. The stereo constraint is checked once all acquisitions have been treated. All constraints must be satisfied in order to obtain a feasible solution. The flowchart of these decoding steps is depicted in Figure 3.

The example of one solution from the modified instance, which needs to schedule two strips, is shown in Figure 4. Two strips are considered in this instance; therefore the number of genes equals 4. The random-keys are generated for all genes and each gene represents one acquisition. The decoding steps are used to obtain the sequence of selected acquisitions and the values of the two objective functions.

3.3 Biased Random-Key Genetic Algorithm

In BRKGA, the new population is combined from three parts (selection, crossover, and mutation) [7]. The first part is the selection part in which we can choose a selection method from several efficient algorithms, e.g., NSGA-II [8], IBEA [9], SMS-EMOA [10], etc. We propose two selection methods to choose p_e preferred chromosomes (elite set) from the current population. We copy these p_e chromosomes to the top part of the next population. The two methods are:

1. **Fast nondominated sorting and crowding distance assignment**
   Fast nondominated sorting and crowding distance assignment methods were proposed in the Nondominated Sorting Genetic Algorithm II (NSGA-II) [8]. In our work, the fast non-dominated sorting method is used to find the solutions in rank zero (nondominated solutions). If the number of nondominated solutions is more than the parameter setting value of maximum size of the elite set, the crowding distance assignment method is applied to select some solutions from the nondominated set to become the elite set. Otherwise all nondominated solutions will become the elite set. The concept of the crowding distance assignment method is to get an estimate of the density of solutions surrounding a particular solution in the population.
2. **Indicator based on the hypervolume concept**
   The use of an indicator based on the hypervolume concept was proposed in the Indicator-Based Evolutionary Algorithm (IBEA) [9]. The indicator based method is used to assign fitness values based on the hypervolume concept to the population members and some solutions in the current population are selected to become the elite set for the next population. The

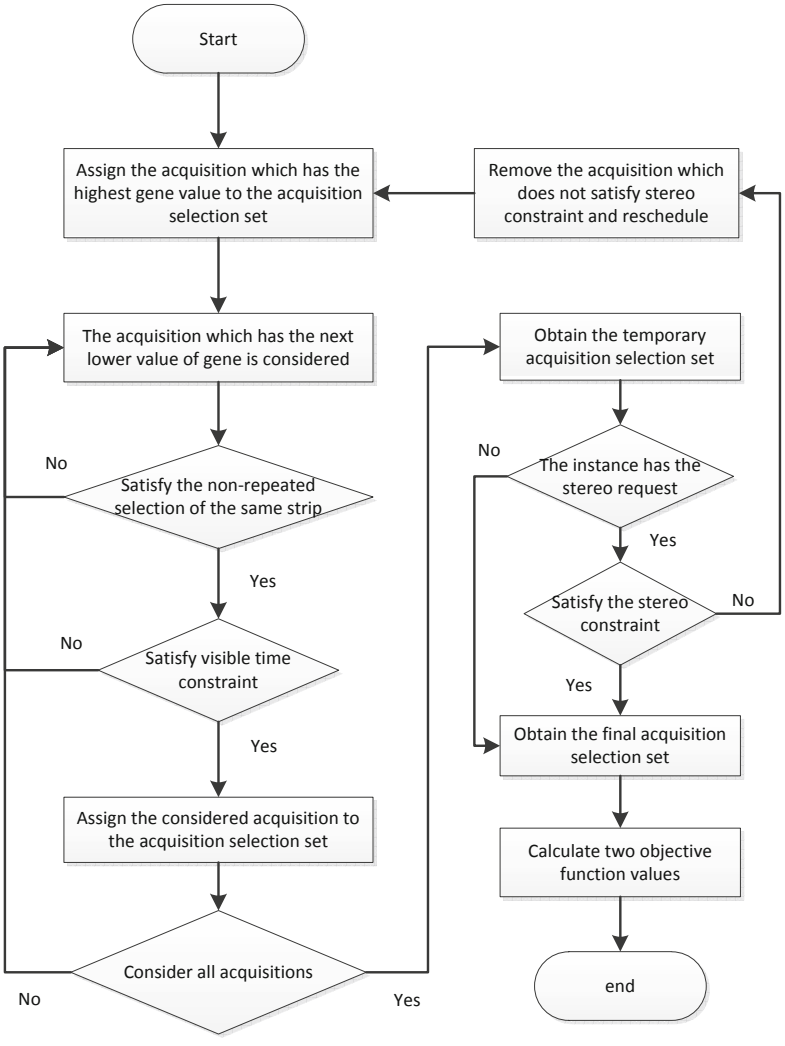

Fig. 3. Decoding steps flowchart of one chromosome into one solution

Random-key chromosome	**Acquisition0**	**Acquisition1**	**Acquisition2**	**Acquisition3**
	0.6984	**0.9939**	**0.6885**	**0.2509**
Sequence of selected acquisitions	**1** | **2**			
Total profit	**1.04234e+007**			
Maximum difference profit	**5.21172e+006**			

Fig. 4. Solution example from the modified instance, which needs to schedule two strips

indicator-based method performs binary tournaments for all solutions in the current population and implements environmental selection by removing the worst solution from the population and by updating the fitness values of the remaining solutions. The worst solution is removed repeatedly until the number of remaining solutions satisfies the recommended size of the elite set for BRKGA.

The second part is the bottom part which is the mutant set. It is the set of p_m chromosomes generated to avoid entrapment in a local optimum. These chromosomes are randomly generated by the same method used to generate the initial population. The last part is the crossover part for which each crossover offspring is built from one elite chromosome and one chromosome in the previous population. Each element in the crossover offspring is obtained from the element in elite chromosome with the probability ρ_e. The crossover offspring is stored in the middle part of the new population. Hence, the size of crossover offspring set is $p - p_e - p_m$ to fulfill the remaining space of chromosomes in the next population. The process for generating the next populations is applied repeatedly until the stopping criterion is satisfied.

4 Computational Results

The ROADEF 2003 challenge instances (subsetA) from ROADEF Challenge website (`http://challenge.roadef.org/2003/en/sujet.php`) 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. For the proposed biased random-key genetic algorithm, the recommended parameter value settings is displayed in Table 1 [7]. Two population sizes of n and $2n$, where n is the length of a chromosome, are tested. The best solutions are stored in the archive set. If there is at least one solution from the current population that can dominate some solutions in the archive set, the archive set will be updated. Thus, we use the number of iterations of the last archive set improvement to be the stopping criterion. The algorithms were experimentally tuned and the stopping value is set to 50. The size of the elite set is equal to the number of non-repeating photograph scheduling results from the nondominated solutions, but it is not over $0.15p$. The size of the mutant set is $0.3p$. The probability of elite element inheritance for crossover operation

Table 1. Recommended parameter values of BRKGA [7]

Parameter	Recommended value
p	$p = a.n$, where $1 \leq a \in \mathbb{R}$ is a constant and n is the length of the chromosome
p_e	$0.10p \leq p_e \leq 0.25p$
p_m	$0.10p \leq p_m \leq 0.30p$
ρ_e	$0.5 \leq \rho_e \leq 0.8$

Table 2. Results of the modified ROADEF 2003 challenge instances subsetA

	Dominance-based				
	Instance	Hypervolume		# solutions	CPU time (s)
		Average	σ		
Population size n	2_0_2	-	-	-	-
	4_0_7	4.6734×10^{15}	8.19837×10^{14}	4.8	0
	12_2_25	5.00258×10^{16}	2.62563×10^{14}	11.1	1611.7
	12_9_28	1.23618×10^{16}	1.06097×10^{15}	7.1	0.3
	68_12_106	2.50509×10^{17}	2.22774×10^{16}	20.8	24.2
	77_40_147	2.69889×10^{16}	3.93096×10^{15}	25.3	77.7
	218_39_295	4.62395×10^{17}	1.11213×10^{17}	25.4	483.9
	150_87_342	4.80749×10^{17}	3.59237×10^{16}	27.7	938.9
	336_55_483	1.24056×10^{18}	1.58973×10^{17}	23.8	2068.2
	375_63_534	1.09387×10^{18}	1.14147×10^{17}	28	1878.9
Population size $2n$	2_0_2	5.43238×10^{13}	0	1	0
	4_0_7	5.78803×10^{15}	6.11588×10^{13}	7.2	0.2
	12_2_25	5.08453×10^{16}	6.69058×10^{13}	23.5	3489.9
	12_9_28	1.24857×10^{16}	1.43998×10^{15}	7.2	0.6
	68_12_106	2.51852×10^{17}	2.20262×10^{16}	37.5	68.8
	77_40_147	2.96205×10^{16}	3.19952×10^{15}	36.4	172.2
	218_39_295	4.50592×10^{17}	5.74569×10^{16}	26.7	937
	150_87_342	5.05842×10^{17}	4.40302×10^{16}	30.2	2013.7
	336_55_483	1.15971×10^{18}	1.58218×10^{17}	27.7	2381.2
	375_63_534	1.24831×10^{18}	1.93988×10^{17}	26.5	4473.1

	Indicator-based				
	Instance	Hypervolume		# solutions	CPU time (s)
		Average	σ		
Population size n	2_0_2	-	-	-	-
	4_0_7	5.36459×10^{15}	2.82301×10^{14}	5.5	0
	12_2_25	4.49038×10^{16}	1.5282×10^{15}	10.2	0.2
	12_9_28	1.16085×10^{16}	1.56771×10^{15}	5.4	0.3
	68_12_106	2.38139×10^{17}	3.06587×10^{16}	9.4	24.3
	77_40_147	2.56001×10^{16}	2.87972×10^{15}	12.5	58.1
	218_39_295	4.97074×10^{17}	6.973×10^{16}	13.3	450.5
	150_87_342	4.4792×10^{17}	2.9806×10^{16}	12.2	852.3
	336_55_483	1.26768×10^{18}	1.07123×10^{17}	10.3	2666.4
	375_63_534	1.34292×10^{18}	1.45157×10^{17}	10.3	4623.4
Population size $2n$	2_0_2	5.43238×10^{13}	0	1	0
	4_0_7	5.81005×10^{15}	5.93069×10^{13}	8.8	0
	12_2_25	4.9165×10^{16}	1.27084×10^{15}	23.1	1.1
	12_9_28	1.24013×10^{16}	9.66474×10^{14}	8.1	1.4
	68_12_106	2.55655×10^{17}	2.32915×10^{16}	13.2	88.4
	77_40_147	2.73546×10^{16}	2.72353×10^{15}	14.7	223.4
	218_39_295	5.30441×10^{17}	3.9777×10^{16}	13.2	3017.3
	150_87_342	4.8244×10^{17}	3.30219×10^{16}	14	5276.6
	336_55_483	1.41375×10^{18}	8.84685×10^{16}	10.3	12904.2
	375_63_534	1.39739×10^{18}	1.07724×10^{17}	13.4	28760

is 0.6. Two methods (dominance-based and indicator-based) for selecting some solutions to become the elite set are tested. They are implemented in C++. We test ten runs per instance. Hypervolumes of the approximate Pareto front are calculated by using a reference point of 0 for the first objective and the maximum of the profit summations of each user for the second one. The average and standard deviations values of hypervolumes, the average number of solutions, and average CPU times are reported in Table 2. Each method obtains a set of solutions, which considers both objective functions (maximize total profit and ensure fairness among users) and all constraints of agile EOSs are satisfied. For both methods, when comparing the results from two different population sizes, most of them show that the methods with the population size $2n$ obtain the better average and standard deviation values of hypervolumes and acquire more solutions, but CPU times are higher. In the other way, when we compare results between dominance-based and indicator-based, the average values of hypervolumes cannot show exactly which one obtains the better solutions. However, the standard deviation for the population size $2n$ of indicator-based is better than dominance-based. On the number of solutions and CPU time, dominance-based obtains more solutions and spends less CPU times, especially for large instances. Except for instance `12_2_25`, dominance-based takes very high CPU time and this is strange. Hence, more tests are done to check the number of iterations until the stopping criterion is satisfied for instance `12_2_25` and instance `12_9_28`. The average number of iterations for instance `12_2_25` and instance `12_9_28` are 2657535.7 and 177.7, respectively. Therefore, instance `12_2_25` spends very high CPU time, because it uses a huge number of iterations until the stopping criterion is satisfied. For instance `2_0_2` when using the population size n, both methods cannot reach any result, because the population size is too small for generating the new generation from 3 parts in BRKGA. Nevertheless, the computation times for large instances are quite high, that means that the efficiency of the decoding methods certainly deserves to be improved.

5 Conclusions and Future Work

Multi-objective optimization is applied to solve the problem of selecting and scheduling the observations of agile Earth observing satellites. The instances of ROADEF 2003 challenge are modified in order to take account explicitly of 4-user requirements. Two objective functions are considered to maximize the total profit and to minimize the maximum difference profit between users for the fairness of resource sharing. Moreover, all constraints have to be satisfied. A biased random-key genetic algorithm (BRKGA) is applied to solve this problem. Random-key encoding generates each chromosome in the population and all of them are decoded to be the solutions. Thus, two methods, fast nondominated sorting with crowding distance assignment on the one hand and indicator based on the hypervolume concept on the other hand, are used for selecting the elite set of solutions from the population. An elite set, a crossover offsprings set, and a mutant set are combined to become the next population. The results of the

dominance-based and indicator-based methods with two population sizes are compared. The approximate solutions are obtained but the computation times for large instances are quite high.

This work is still in progress. As a future work, we plan to use other random-key decoding methods in order to reduce the computation times. Moreover, we will apply indicator-based multi-objective local search (IBMOLS) to solve this problem and compare the IBMOLS results with the BRKGA results which are proposed in this paper.

Acknowledgment. This work was supported by THEOS Operational Training Programme (TOTP). The authors gratefully acknowledge this support. Thanks are also due to the referees for their valuable comments.

References

1. Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.M., Bataille, N.: Selecting and Scheduling Observations of Agile Satellites. Aerospace Science and Technology 6, 367–381 (2002)
2. Kuipers, E.J.: 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 (2003)
3. Cordeau, J.F., Laporte, G.: Maximizing the Value of an Earth Observation Satellite Orbit. Journal of the Operational Research Society 56, 962–968 (2005)
4. Bataille, N., Lemaître, M., Verfaillie, G.: Efficiency and Fairness when Sharing the Use of a Satellite. In: Proc. Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 465–470 (1999)
5. Bianchessi, N., Cordeau, J.F., Desrosiers, J., Laporte, G., Raymond, V.: A Heuristic for the Multi-Satellite, Multi-Orbit and Multi-User Management of Earth Observation Satellites. European Journal of Operational Research 177, 750–762 (2007)
6. Verfaillie, G., Lemaître, M., Bataille, N., Lachiver, J.M.: Management of the Mission of Earth Observation Satellites Challenge Description. Technical report, Centre National d'Etudes Spatiales, France (2002)
7. Gonçalves, J.F., Resende, M.G.C.: Biased Random-Key Genetic Algorithms for Combinatorial Optimization. Journal of Heuristics 17, 487–525 (2011)
8. Deb, K., Pratep, A., Agarwal, S., Meyarivan, T.: A Fast and Elite Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
9. Zitzler, E., Künzli, S.: 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 (2004)
10. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181, 1653–1669 (2007)