

Genetic Algorithm and Local Search Comparison for Solving Bi-objective p-Median Problem

Panwadee Tangpattanakul
 Geo-Informatics and Space Technology Development Agency
 (Public Organization)
 120, The Government Complex (Building B),
 Chaeng Wattana Road, Laksi, Bangkok 10210, Thailand
 Email: panwadee@gistda.or.th

Abstract—This paper presents two algorithms, which are a nondominated sorting genetic algorithm II (NSGA-II) and an indicator-based multi-objective local search (IBMOLS), for solving a bi-objective p-Median problem. The bi-objective p-Median problem is a problem of finding p location points to install facilities from a set of m candidates. This problem considers two objectives: minimizing the sum of the distances from each customer to the nearest facility and minimizing the sum of the costs to install each facility in the selected location points. NSGA-II and IBMOLS are efficient algorithms in the area of multi-objective optimization. Experiments are conducted on generated instances. Hypervolume values of the approximate Pareto fronts are computed and the obtained results from IBMOLS and NSGA-II are compared.

I. INTRODUCTION

This paper proposes two efficient multi-objective optimization algorithms, which are a nondominated sorting genetic algorithm II (NSGA-II) and an indicator-based multi-objective local search (IBMOLS), to solve the bi-objective p-Median problem. We experiment on instances, which consider different numbers of candidates and numbers of facilities.

The p-Median problem is a network-based location problem in the area of location analysis [1], [2]. It needs to find p locals, which use to install facilities, from a set of m candidates. This problem is an optimization problem that needs to reach the objective functions and satisfy the constraints. The p-Median problem is classified as NP-hard problem [3]. Generally, the p-Median problem considers only one objective, which is to minimize the sum of the distances from each customer to the nearest facility. However, two objectives are considered simultaneously in this paper. The second objective is to minimize the sum of the costs to install each facility in the selected location point. Thus, the multi-objective optimization algorithms can be applied to solve the considered problem.

The multi-objective optimization is an interesting area among researchers until nowadays. It has been continuously developed since the multi-objective optimization can be formulated to assimilate real world questions. However, the methods for solving multi-objective optimization problems are much more complicated, and they require efficient optimization strategies. Several heuristic methods have good performance for solving this type of problems. Therefore, heuristic methods, which are a nondominated sorting genetic algorithm (NSGA-

II) and an indicator-based multi-objective local search (IBMOLS), are used to solve the considered problem in this work and both results are compared. In the recent years, the single objective p-Median problem was solved by using the genetic algorithms, and it obtained good solutions [4], [5]. For the multi-objective p-Median problem, a multi-objective genetic algorithm (MOGA) [6] and an iterated local search heuristic (ILS) [7] were proposed for solving the problem.

The article is organized as follows. Section II explains the multi-objective p-Median problem. The formulation of the considered problem and the difficulty of the multi-objective optimization problem are presented. The strategies of NSGA-II and IBMOLS are presented in Section III. Section IV reports the computational results. Finally, conclusions is discussed in Section V.

II. MULTI-OBJECTIVE P-MEDIAN PROBLEM

Generally, the p-Median problem considers only one objective, which is to minimize the sum of the distance from each customer to the nearest facility. However, the second objective is added in our problem to minimize the sum of the costs to install each facility in the selected location point. Thus, the formulation of this problem is as follows [6]:

$$\begin{aligned}
 & \text{minimize} && \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \\
 & \text{minimize} && \sum_{j \in J} f_j y_j \\
 & \text{subject to} && \sum_{j \in J} x_{ij} = 1, \quad i \in I \\
 & && x_{ij} \leq y_j, \quad i \in I, j \in J \\
 & && \sum_{j \in J} y_j = p \\
 & && x_{ij}, y_j \in \{0, 1\}, \quad i \in I, j \in J.
 \end{aligned}$$

The values of d_{ij} is the distance from customer i to facility at location point j , and f_j is the cost to install the facility at location point j . The decision variable x_{ij} equals to 1, if customer i is serviced by facility at location point j , otherwise, x_{ij} equals to 0. The decision variable y_j equals to 1, if the facility is installed at location point j , otherwise y_j equals to 0.

The first constraint guarantees that each customer is serviced by only one installed facility and each customer is supplied from an facility at the installed location points in the second constraint. The number of the installed localtion points is equal to p by the third constraint.

The fitness of each individual must be taken into account for all objective functions; therefore, we will not obtain only one optimal solution, but a set of solutions. The obtained tradeoff solutions are called Pareto optimal solutions. Because of this, the algorithms that use to solve the multi-objective optimization problem are more complicated than the algorithms for solving the single objective optimization problem. In this work, algorithms for solving the multi-objective optimization problem are studied. The nondominated sorting genetic algorithm II (NSGA-II) and the indicator-based multi-objective local search (IBMOLS) are proposed to solve the considered problem.

III. GENETIC ALGORITHM AND LOCAL SEARCH STRATEGIES

In this section, two methods, which are applied to solve the multi-objective p-Median problem, are explained. The first methods is a nondominated sorting genetic algorithm II (NSGA-II). It is an efficient and well-known algorithm in the area of multi-objective optimization. The second method is an indicator-based multi-objective local search (IBMOLS). It is a generic algorithm, which combine the use of basic local search and binary indicator.

A. Nondominated sorting genetic algorithm II

Nondominated sorting genetic algorithm II or NSGA-II was proposed in [8]. The genetic algorithm is a population-based metaheuristic method which is developed from the natural survival concept. It operates by using several individuals in the population. Each individual consists of a chromosome, which represents one solution of the search space. For NSGA-II, the individuals in the population are classified into several layers of classification. NSGA-II uses the dominance relation in the Pareto sense for ranking the individuals. This is called the fast nondominated sorting. All nondominated individuals are classified in the first rank. The ranking classification process continues until all individuals in the population are classified. Then, NSGA-II uses the crowding distance assignment for ensuring the diversity in the considered front. The crowding distance assignment sorts the individuals depending on the distance between the two adjacent points in the objective space. The individuals are ordered in the sorted set according to the distance values from high to low. For the boundary points, they always stay in the first positions of the sorted set, so that they are always selected.

Each chromosome represents one solution. In this work, the size of each chromosome is set equal to the value of p (number of facilities that need to be installed). Each gene represents the location point, which is selected to install the facility. Thus, the gene value is an integer in the interval $[1, m]$, where m is the number of candidate location points.

In [9], two methods were used to generate the initial chromosomes for the first iteration in the NSGA-II process. The two methods were a random generation and a distributed generation. The random generation assigned randomly the gene values of each chromosome. The size of population (p_o) was fixed and it was set to 200. For the distributed generation, it was borrowed from [4]. The size of population was computed by using the idea that the population size should be proportional to the number of feasible solutions. All candidates must be assigned in the genes of at less two chromosomes in the population. In the computational results of [9], the random generation obtained better solutions than the distributed generation. Thus, we use the random generation to generate the initial population for the NSGA-II process in this work.

To generate offspring of NSGA-II, the selection, crossover and mutation operators are used. For the selection step, the p_e offspring are generated. To generate each offspring, two parents are randomly chosen from the population, and the tournament selection is applied. The winner from the tournament selection becomes one offspring in the selection set. For the crossover process, the number of crossover offspring is set to p_c . A biased gene values assignment, which is borrowed from [10], is applied. Each offspring is built by using the crossover operator between a randomly selected chromosome from the nondominated set and another randomly selected chromosome from the whole current population. Each element in the offspring is obtained from the element of nondominated parent with the probability ρ_e . Otherwise, the element of offspring is copied from the other parent. For the mutation step, the offspring is generated by using the random generation. The number of mutation offspring is equal to p_m . The offspring set and the current population are combined. Then, the fast nondominated sorting and the crowding distance assignment are used for selecting p_o best fitness individuals to become the new population. The generations are repeated until a stopping criterion is satisfied.

B. Indicator-based multi-objective local search

The indicator-based multi-objective local search or IBMOLS was initially proposed in [11]. It is an iterated local search, which generates an approximate Pareto front PO as an empty set before starting the first iteration and this set is updated at the end of each iteration. IBMOLS starts by generating the initial population for each iteration. In this work, the random generation is used for the first iteration and the perturbation is used for the other iterations. Then, the nondominated solutions from the population are stored in the archive set A . In [11], two indicators, an epsilon indicator and a hypervolume indicator, were proposed to be the quality indicators of the individuals in the population. We choose to used the hypervolume indicator in this work. The fitness values of all individuals in the population are computed by using the indicator based on the hypervolume concept from the indicator-based evolutionary algorithm or IBEA in [12]. The hypervolume is an indicator, which measures the area or

volume dominated by the set of considered solutions. The local search step is applied for each individual until all individuals are treated. The updated population 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 in this step, the process will return to apply the local search step again. 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, the perturbation is used to generate a new initial population for the next iteration of IBMOLS.

1) *Initial population generation:*

- *Random generation for the first iteration*

A random generation is applied to generate the initial population for the first iteration. Each individual in the population has a chromosome, which represents the selected p location points for the facilities installation. Genes in the chromosome are generated randomly in the interval $[1, m]$ and each gene cannot be repeated.

- *Perturbation for the other iterations*

For the other iterations, the perturbation is applied to generate the initial population. The original chromosome is randomly selected from the approximate Pareto front PO . Then, some genes in the original chromosome is replaced by some non-selected location points. The number of replacing genes is set to 0.2 of the chromosome size. Then, the replaced chromosome becomes a chromosome in the initial population of the next iteration.

2) *Local search step:* Each individual in the population is considered one by one and its neighborhood is explored by replacing a gene in the considered chromosome. The generated neighbor is included in the population and the fitness of each individuals in the population is updated. Then, the worst individual is removed from the population and the fitness of the remaining individuals are updated. If the included neighbor is the same individual as the worst one, the new neighbor will be re-generated by exploring the neighborhood.

The neighborhood of all individuals in the population are explored and the population is updated. The archive set A is also updated by storing the nondominated solutions of the combined set, which combines the updated population and the previous iteration archive set A . If the updated archive set A does not change, the local search step will be stopped. Otherwise, another local search step is performed.

The neighbor is generated by replacing a gene in the chromosome with a non-selected location point. The position of replacement is randomly chosen and the replaced non-selected location point is also random selected. The neighbor will be re-generated, if the previous included neighbor is the same individual as the worst individual and all neighbors in the neighborhood is not explored.

IV. COMPUTATIONAL RESULTS

The proposed algorithms are tested on sixteen randomly generated instances. The format of instance names is set to Instance m_p where m is the number of candidate location points, and p is the number of selected points.

The parameter values of the two proposed algorithms are experimental tuned. For NSGA-II, the population size is set depending on the initial population generation method. In this paper, we use the random generation for generating the initial population. Thus, the population is fixed and set to 200. The size of offspring sets from selection, crossover, and mutation are equal to 0.3, 0.4, and 0.3 of the population size, respectively.

For the crossover process, two parents are random selected from the population. The first one is from the nondominated set and the second one is from the entire population. Each element of the offspring is from the first parent with a probability of 0.7 and from the second parent with a probability of 0.3. The number of iterations since the last hypervolume values improvement is chosen to be a stopping criterion, and it is set to 500 iterations. It means that the algorithm will terminate, if hypervolume value improvement is stable for more than 500 iterations. For IBMOLS, the population size is set to 10. The random generation is applied to generate the initial population for the first iteration and the perturbation is applied to generate the initial population for the other iterations. In the perturbation process, some genes in the chromosome are replaced by the non-selected location points. The number of the replaced genes is set to 0.2 of the chromosome size. The number of iterations since the last hypervolume values improvement is also chosen to be a stopping criterion, and it is set to 10 iterations. According to the IBMOLS procedure, there is two other steps, which are the population development checking and the archive set change checking for operating the repeated process, thus the stopping value does not need to be set as high as in NSGA-II.

The objective function values are normalized to a range between 0 and 1, by dividing with the sum of the distances from each customer to the farthest facility for the first objective and the sum of p maximum costs to install the facilities for the second objective. In each iteration, the hypervolume of the nondominated solutions set are computed. To ensure that all solutions in the nondominated solutions set, even the solutions obtaining the worst objective values, are considered for computing the hypervolume value, thus the reference point of the hypervolume is set to (2,2).

The algorithms are implemented in C++, and thirty runs per instance are tested. Two proposed methods are applied to solve the multi-objective p-Median problem. The set of testing instances consists of sixteen instances, which consider different numbers of candidate location points (m) and different numbers of selected points (p). The results which are obtained from the two methods are compared. For thirty runs, the hypervolume values of the approximate Pareto front are computed. Box plots of all instances, which illustrate the

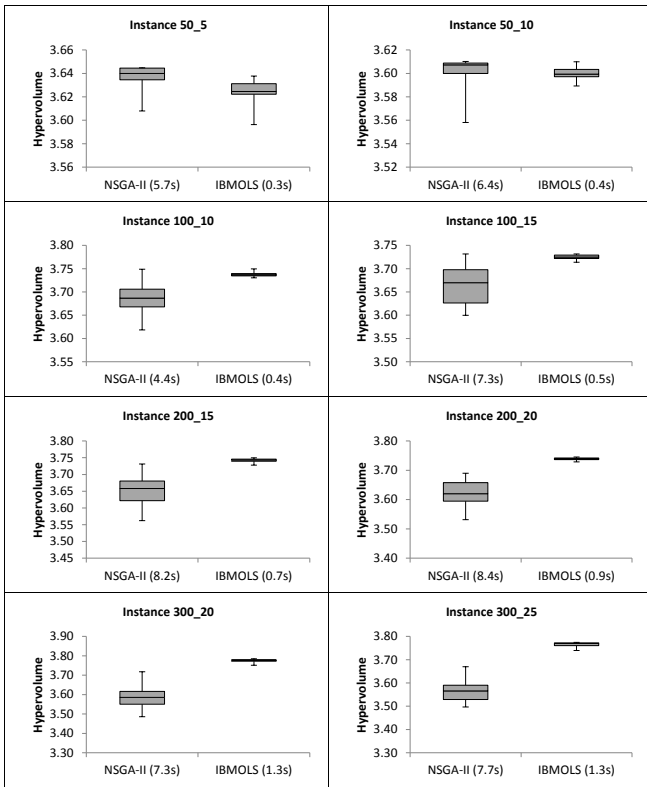


Fig. 1. Comparison of the hypervolume values of the approximate Pareto front and the average computation times between NSGA-II and IBMOLS for small and medium instances

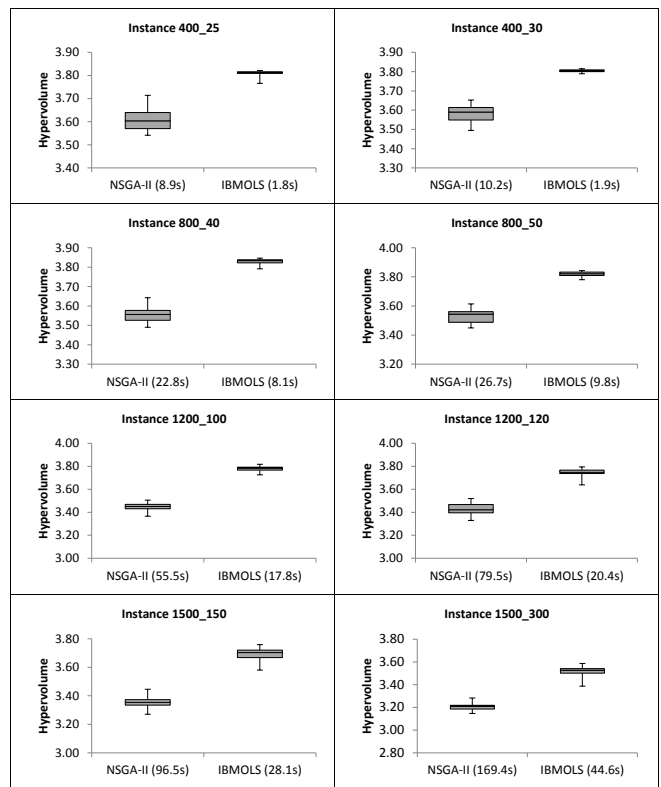


Fig. 2. Comparison of the hypervolume values of the approximate Pareto front and the average computation times between NSGA-II and IBMOLS for large instances

maximum value, the median value, the minimum value, and the interquartile range of the hypervolume values, and the average computation times, are presented in Figure 1 and Figure 2.

The box plots show that NSGA-II obtains better hypervolume values for the small instances, which consider 50 candidate location points (Instance 50_5 and Instance 50_10). However, IBMOLS obtains better hypervolume values for all other instances, both on median value and standard variation. Moreover, it spends less computational times. We notice that IBMOLS obtains better solutions, even if the population size and the stopping criterion value for IBMOLS are set much less than for NSGA-II. IBMOLS is more efficient than NSGA-II for the medium and large instances, since IBMOLS uses the easiness of the basic local search, the advantage of population-based algorithm, which explores diverse initial solutions in the solution space, and it also uses the efficiency of the binary-indicator for selecting the good solutions. IBMOLS can obtain better approximate Pareto front and the best approximate Pareto fronts of all tested instances are illustrated in Figure 3 and Figure 4. The figures of the approximate Pareto front show that the multi-objective optimization algorithms give many nondominated solutions on the approximate Pareto front. Then, the decision makers can choose one preferred solution from the approximate Pareto front to be the best solution of this problem.

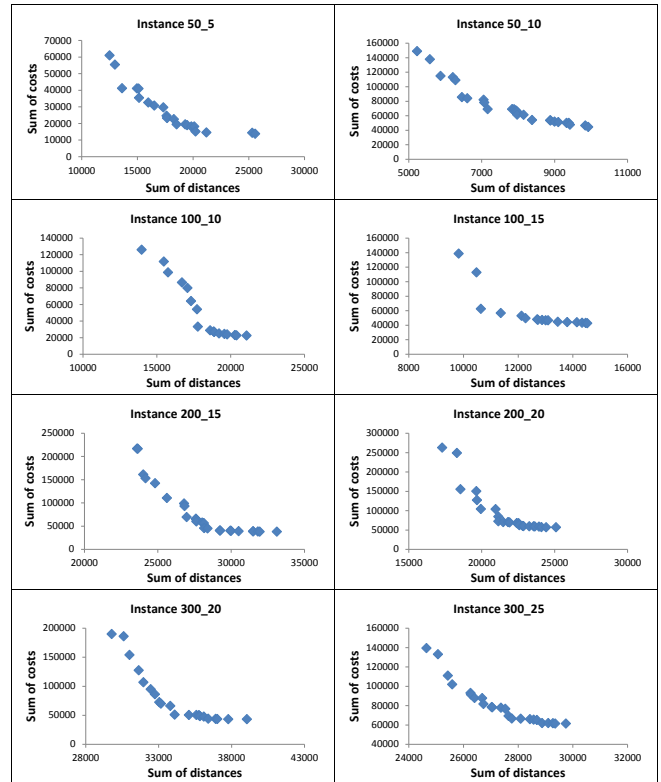


Fig. 3. The best approximate Pareto fronts, which are obtained from IBMOLS, for small and medium instances

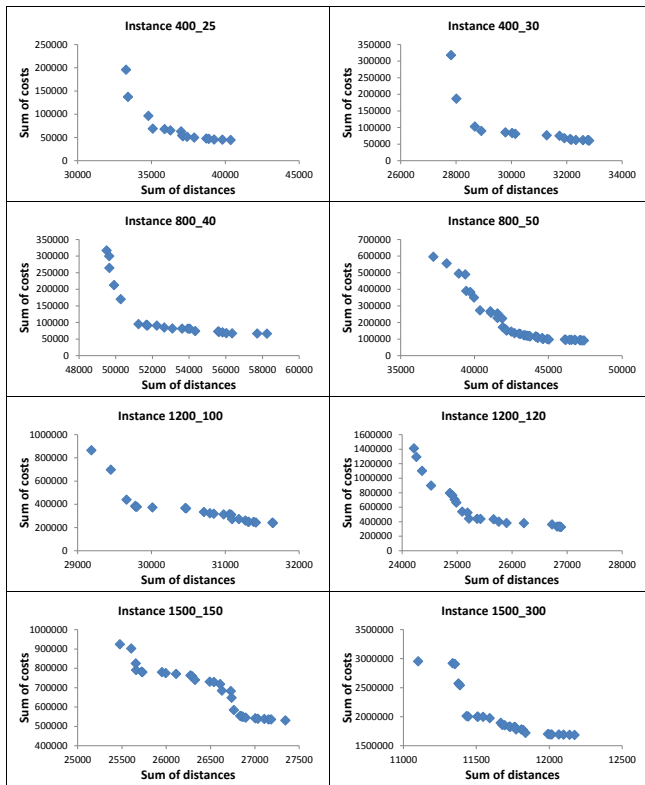


Fig. 4. The best approximate Pareto fronts, which are obtained from IBMOLS, for large instances

V. CONCLUSIONS AND FUTURE WORKS

A nondominated sorting genetic algorithm (NSGA-II) and an indicator-based multi-objective local search (IBMOLS) are applied to solve a multi-objective p -Median problem. NSGA-II uses many chromosomes in the population. Each chromosome consists of several genes. Each gene represents one selected location point, which is assigned in the solution. In this work, the random generation is used to generate the initial population. In the process of NSGA-II, the selection, crossover, and mutation operators are applied. A binary tournament is used in the selection process; a biased gene value assignment is applied for the crossover; and the random generation is also used in the mutation step. For IBMOLS, the random generation and the perturbation are used to generate the initial population for the first iteration and the other iterations, respectively. The nondominated solutions in the population are stored in the archive set and the approximate Pareto front is updated, when each iteration finishes. The experiments are conducted on sixteen randomly generated instances. The objectives of this problem are to minimize the sum of the distances from each customer to the nearest facility and to minimize the sum of the costs to install each facility in the selected location points. The results, which are obtained from NSGA-II and IBMOLS, are compared.

NSGA-II obtains better hypervolume value for the small instances. However, for the medium and large instances,

IBMOLS obtains better solutions, both on median value and standard variation. Moreover, it spends less computation times than NSGA-II.

REFERENCES

- [1] C. S. ReVelle and H. A. Eiselt, *Location analysis: A synthesis and survey*, European Journal of Operational Research, 165, 1-19, 2005.
- [2] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama, *Facility location and supply chain management - A review*, European Journal of Operational Research, 196, 401-412, 2009.
- [3] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, *The p -median problem: A survey of metaheuristic approaches*, European Journal of Operational Research, 179, 927-939, 2007.
- [4] O. Alp, E. Erkut, and Z. Drezner, *An efficient genetic algorithm for the p -Median problem*, Annals of Operations Research, 122, 21-42, 2003.
- [5] E. S. Correa, M. Teresinha, A. Steiner, A. A. Freitas, C. D. Parana, and C. Carnieri, *A genetic algorithm for the p -median problem*, In: Proceedings of 2001 Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 1268-1275, 2001.
- [6] J. E. C. Arroyo, P. M. dos Santos, M. S. Soares, and A. G. Santos, *A multi-objective genetic algorithm with path relinking for the p -Median problem*, In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. Lecture Notes in Artificial Intelligence (LNAI), vol. 6433, pp. 70-79, Springer, Heidelberg, 2010.
- [7] J. E. C. Arroyo, A. G. Santos, P. M. dos Santos, and W. G. Ribeiro, *A bi-objective iterated local search heuristic with path relinking for the p -Median problem*, In: R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, (eds.) EMO 2011. Lecture Notes in Computer Science (LNCS), vol. 6576, pp. 492-504, Springer, Heidelberg, 2011.
- [8] K. Deb, A. Pratep, S. Agarwal, and T. Meyarivan, *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6, 182-197, 2002.
- [9] P. Tangpattanukul, *Genetic Algorithm for Multi-objective p -Median Problem*, In: Proceedings of the 37th Electrical Engineering Conference, Khon Kaen, Thailand, 2014.
- [10] J. F. Gonçalves and M. G. C. Resende, *Biased random-key genetic algorithms for combinatorial optimization*, Journal of Heuristics, 17(5), 487-525, 2011.
- [11] M. Basseur and E. K. Burke, *Indicator-Based Multi-Objective Local Search*, In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 3100-3107, Singapore, 2007.
- [12] E. Zitzler and S. Künzli, *Indicator-Based Selection in Multi-objective Search*, In: X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, and H.-P. Schwefel, (eds.) PPSN VIII. Lecture Notes in Computer Science (LNCS), vol. 3242, pp. 832-842, Springer, Heidelberg, 2004.