Chapter 5
Greedy Randomized Adaptive Search Procedure

Sergio Pérez-Peló, Jesús Sánchez-Oro, and Abraham Duarte

5.1 Greedy Randomized Adaptive Search Procedure

Greedy randomized adaptive search procedure (GRASP) is a trajectory-based metaheuristic originally proposed in Feo and Resende (1989) for solving the set covering problem and then formally defined in Feo et al. (1994). GRASP follows a multi-start approach, dividing each iteration into two different processes. The first process, named construction, consists of generating a feasible solution for a given problem from scratch, while the second process, named local improvement, is responsible for finding a local optimum with respect to the constructed solution. We refer the reader to Festa and Resende (2009a,b) for a detailed survey on recent advances on GRASP and a detailed description of the main variants. Algorithm 5.1 shows the pseudocode for the complete GRASP algorithm.

The method requires from three input parameters: V, the set of available elements in the context of a diversity problem; α, which is a parameter of the constructive phase which will be later discussed; and γ, the number of complete iterations of GRASP to be performed. As a trajectory-based metaheuristic, GRASP maintains only the best generated solution among all iterations (step 1). Then, the method performs γ complete iterations (steps 2–8). In each iteration, a feasible solution M is constructed with a greedy randomized constructive procedure GRC (step 3). Then, the generated solution is locally optimized with a local improvement method LI (step 4) which is usually a local search method, generating a local optimum M'. Once a complete iteration has been performed, the method evaluates if the new solution M' outperfoms the best solution found so far M_b (step 5), updating the best

S. Pérez-Peló · J. Sánchez-Oro (✉) · A. Duarte
Department of Computer Sciences, Universidad Rey Juan Carlos, Mostoles, Spain
e-mail: sergio.perez.pelo@urjc.es; jesus.sanchezoro@urjc.es; abraham.duarte@urjc.es

R. Martí, A. Martínez-Gavara (eds.), *Discrete Diversity and Dispersion Maximization*, Springer Optimization and Its Applications 204,
https://doi.org/10.1007/978-3-031-38310-6_5

Algorithm 5.1 $GRASP(V, \alpha, \gamma)$

1: $M_b \leftarrow \emptyset$
2: **for** $i \in 1 \ldots \gamma$ **do**
3: $\quad M \leftarrow GRC(V, \alpha)$
4: $\quad M' \leftarrow LI(M)$
5: $\quad$ **if** $f(M') > f(M_b)$ **then**
6: $\qquad M_b \leftarrow M'$
7: $\quad$ **end if**
8: **end for**
9: **return** M_b

solution found if necessary (step 6). The method ends returning the best solution found after performing γ iterations (step 9).

One of the most representative phases of GRASP is the greedy randomized constructive (GRC) procedure, which is responsible for the diversity of the metaheuristic. The GRC usually follows a traditional greedy procedure scheme but, instead of selecting the best candidate in each step, GRC selects one of the most promising candidates to be included in the solution, with the aim of generating diverse solutions without deteriorating their quality. Algorithm 5.2 shows the pseudocode of the standard GRC.

Algorithm 5.2 $GRC(V, \alpha)$

1: $v \leftarrow RND(V)$
2: $M \leftarrow \{v\}$
3: $CL \leftarrow V \setminus \{v\}$
4: **while not feasible**(M) **do**
5: $\quad g_{\min} \leftarrow \min_{c \in CL} g(c)$
6: $\quad g_{\max} \leftarrow \max_{c \in CL} g(c)$
7: $\quad \mu \leftarrow g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$
8: $\quad RCL \leftarrow \{c \in CL : g(c) \geq \mu\}$
9: $\quad e \leftarrow RND(RCL)$
10: $\quad M \leftarrow M \cup \{e\}$
11: $\quad CL \leftarrow CL \setminus \{e\}$
12: **end while**
13: **return** M

The algorithm requires from two input parameters: V, which represents the set of points that can be selected, and α, which is a parameter that controls the randomness of the method.

With the aim of favor diversity, the method starts by selecting the first element at random (step 1), including it in the solution under construction (step 2). Then, the candidate list (CL) is created with all the elements that can be eventually included in the solution (step 3). The method then iterates until generating a feasible solution (steps 4–12). It is worth mentioning that the definition of feasibility strictly depends on the problem being tackled.

As a greedy procedure, the constructive method requires from a greedy function $g(v)$ which, given an element v, evaluates its contribution to the solution under construction. Since the candidates are evaluated in each iteration, it is recommended to consider an efficient greedy function which does not require high computational effort. In some cases, even the objective function can be used as greedy function. In each iteration the minimum ($g_{\min}$) and maximum ($g_{\max}$) values for this greedy function are computed (steps 5–6). Those values are then used to establish a threshold μ (step 7). Then, the restricted candidate list (RCL) is created with those elements whose greedy function value is better than the threshold μ (step 8). Without loss of generality, the pseudocode considers a maximization problem. When facing minimization problems, it is only required to replace $\geq$ with $\leq$ in step 8 to include in the RCL those elements whose greedy function value is smaller than or equal to the threshold and, additionally, replace symbol $-$ with $+$ in step 7. Then, the RCL contains the most promising elements to be included in the solution under construction. The next element is therefore selected at random from the RCL to favor diversity (step 9), including it in the solution (step 10) and removing it from the candidate list (step 11). The method ends when reaching a feasible solution, returning the constructed solution (step 13).

The threshold that limits the elements included in the restricted candidate list strongly depends on the value of α, which is an input parameter of the constructive procedure. The domain of this parameter is in the range [0,1] and it is responsible for controlling the randomness and greediness of the method. In particular, when considering $\alpha = 1$, the threshold takes the value of $g_{\min}$, resulting in a totally random method, since all the candidates always have a greedy function value larger than or equal to $g_{\min}$. On the contrary, when $\alpha = 0$, the threshold takes the value of $g_{\max}$, becoming a totally greedy method since the only candidates whose greedy function value is larger than or equal to $g_{\max}$ are those whose greedy function value is exactly $g_{\max}$. Therefore, the value of α must be experimentally set depending on the necessity of diversification and intensification of the problem under consideration. Figure 5.1 illustrates this behavior.

In the graphical example, the elements entering in the RCL are limited by a dashed line, which depends on the value of α: the line will move to the left when increasing the value and to the right if it gets decreased. This is the *randomized* phase of GRASP which allows the method to generate different solutions.

The solution generated with GRC in the first phase of GRASP is not necessarily a local optimum with respect to any neighborhood, mainly due to the randomization included in the construction with the aim of increasing the diversity of the search. Therefore, the second phase of GRASP consists on a local improvement that guides the initial solution to a local optimum with respect to a certain neighborhood that strongly depends on the problem being considered. This local improvement phase is usually accomplished by a local search method, but more complex heuristics or even a complete metaheuristic can be considered to further improve the quality of the solutions found. In traditional GRASP designs, it is customary to consider a simple yet effective local search method to reduce the computational effort required to find

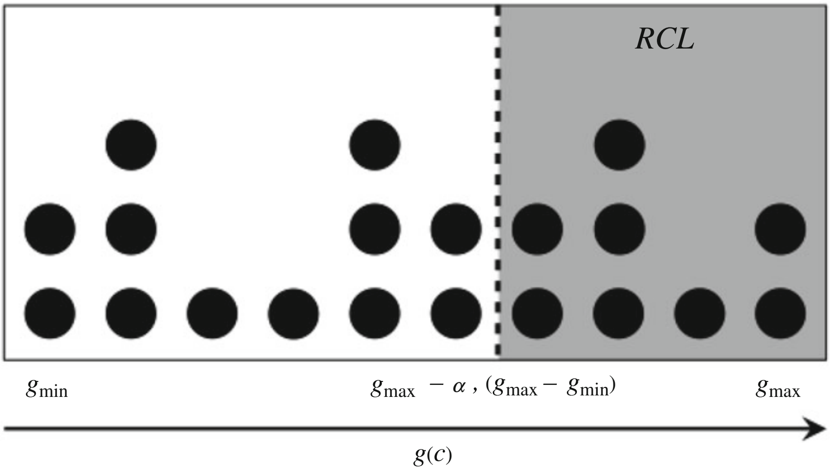

Fig. 5.1 Process of candidate selection depending on the value of α

a local optimum, since the local search will be performed over every constructed solution.

Several alternatives to the GRC have been proposed, but one of the most extended variants of GRC considers the interchange of the random and greedy phases. Specifically, instead of greedily selecting the most promising elements and then randomly selecting the next one, the method randomly selects a set of elements to be selected and then it selects the one with the largest greedy function value among them.

The local search explores a certain neighborhood, which strictly depends on the problem under evaluation. A neighborhood is conformed with a set of solutions that can be reached by performing a single movement operator of the incumbent solution. There are two main strategies to traverse the neighborhood which are commonly used: first improvement and best improvement. Both strategies differ in which movements are accepted in each step of the search. In a first improvement approach, the search performs the first movement that leads to an improved solution, while in best improvement the search performs the movement that results in the best solution of the explored neighborhood. Notice that when considering a first improvement approach, the order in which the neighborhood is explored is relevant, since it determines the direction followed by the search. We refer the reader to a recent empirical study Hansen and Mladenović (2006) comparing the effect of considering both strategies in the local search phase.

Although GRASP presents a general scheme that can be easily adapted to several hard optimization problems, there are some steps that must be specifically defined for each problem in order to generate high-quality solutions. In particular, the greedy function value of the GRC and the local search phase are completely dependent on the problem under consideration. Section 5.2 presents the most recent advances in GRASP design for diversity problems and Section 5.3 proposes a simple yet effective GRASP algorithm specifically designed for solving diversity problems.

5.2 A Review on GRASP for Solving Diversity Problems

Since its proposal in 1989, GRASP has been continuously evolving and it has attracted the attention of the scientific community, mainly due to its simplicity and versatility to adapt to any kind of hard optimization problem. Although it was originally proposed for solving the set covering problem by Feo and Resende (1989), GRASP has been widely studied in the context of diversity.

Even more, one of the first research works related to diversity, tackled by Hart and Shogan (1987), which was published 10 years before the original GRASP research article, proposed a semi-greedy heuristic for the MaxMin diversity problem. The research introduces randomization in the selection of the next candidate to be selected with the aim of generating different solutions, resulting in better result than a traditional greedy heuristic. This idea inspired the design of GRASP. Additionally, they proposed a local search heuristic based on replacing an element with a new one, stopping when no improvement is found. This local search has been later applied for most of the diversity variants in the literature, being refined in each new proposal.

The $\mathcal{NP}$-hardness of the MaxMin problem was proven by Ghosh (1996). In this work, the author proposed a greedy randomized heuristic which is considered an intermediate phase between simple heuristics and complex metaheuristics.

Several heuristics based on the GRASP framework were proposed for solving the MaxSum model . In their research, Silva et al. (2004) evaluated the combination of different constructive procedures coupled with the local search method, testing then in a wide set of instances. The proposed algorithm was named KLD, and it requires several hours of computing time in instances with 500 elements. A hybrid method named GRASP-DM was then presented by Santos et al. (2005), introducing some data mining techniques in the context of GRASP. GRASP-DM firstly executes the GRASP algorithm a fixed number of iterations and, then, the data mining process extracts patterns from the most promising solutions. After that, a hybrid algorithm combining GRASP with path relinking was proposed by Silva et al. (2007), where GRASP is applied to generate a set of elite solutions and, then, the path relinking method is applied to create a path between one of the elite solutions and a newly generated solution using GRASP.

The combination of GRASP with Tabu Search was explored for solving the MaxSum model by Duarte and Martí (2007), proposing two constructive procedures and a tabu search improvement phase. Authors focused on proposing new strategies for exploring the interchange neighborhood efficiently, to avoid large computing times as in previous works. This is accomplished by selecting only the most promising movement on the neighborhood, instead of exploring the complete one.

The MaxMin model was tackled following a GRASP approach by Resende et al. (2010), using an efficient implementation which is able to reach the best results in the literature at that time in short computing times. The approach considers the distance of the next candidate to the elements already selected as greedy function of GRASP. With the aim of efficiently exploring the flat landscape of the MaxMin problem, authors proposed the use of an alternative objective function which generates a non-flat landscape. The GRASP procedure is coupled with a path relinking strategy to further improve the generated solutions.

The first public dataset of instances for diversity, named MDPLIB, was originally presented by Martí et al. (2013), collecting 315 instances from different previous works. In that survey, authors state that GRASP is one of the most promising metaheuristics for solving diversity problems.

A GRASP algorithm for the MaxMean model, also known as equitable dispersion problem, was proposed by Martí and Sandoya (2013). In this variant, the number of elements to be selected is not fixed and the distances between elements can take either positive or negative values. Authors proposed a GRASP constructive algorithm which considers a novel combination of greediness and randomization, together with a local search method based on variable neighborhood descent and path relinking as a post-processing approach.

Then, Martínez-Gavara et al. (2017) integrate GRASP with Tabu Search in an algorithm which is able to oscillate in the feasibility boundary defined by the MaxMinSum constraint, testing six GRASP variants and three oscillation variants, comparing the results with LocalSolver, a black-box local-search solver for general 0-1 programming proposed by Benoist et al. (2011).

Considering the MinDiff model, Duarte et al. (2015) proposed a GRASP algorithm coupled with exterior path relinking, which is applied to increase the diversity of the search by generating new solutions which are different from both the initial and the guiding solution. The results are compared with the optimal values obtained by the CPLEX commercial solver, showing the effectiveness of the proposal.

Peiró et al. (2021) introduce capacity constraints in the diversity problem. They proposed a hybrid GRASP with the variable neighborhood descent algorithm for the capacitated dispersion problem. Later on, Martí et al. (2021) proposed a new mathematical model and a scatter search algorithm, whose diversification generation method is based on GRASP, improving the best previous results.

5.3 GRASP Design for Solving Maximum Diversity Problem

In this section, a simple yet effective GRASP approach for solving the MaxSum diversity problem (MDP) is proposed. Given a set of V elements, with $|N| = n$, a solution for the MDP, conformed with a subset $M \subseteq V$ of m elements, with $m < n$, is evaluated as the sum of distances among the selected elements. In mathematical terms,

$$MDP(M) = \sum_{i<j, i,j \in M} d_{ij}$$

where d_{ij} is the considered metric of distance between elements i and j, which strictly depends on the instance under evaluation. Then, the objective of MDP is to find a solution $M^\star$ with the maximum objective function value among all possible solutions:

$$S^\star \leftarrow \arg\max_{M \in \mathbb{M}} MDP(M)$$

where $\mathbb{M}$ represents all possible subsets of m elements that can be conformed with the elements in V.

In order to show the appropriateness of GRASP for solving diversity problems, we propose a simple yet effective algorithm conformed with a greedy randomized and adaptive constructive procedure coupled with a local search method.

The constructive procedure follows the traditional GRC scheme proposed in Section 5.1. The greedy function used in this case is the actual contribution of the candidate to be selected to the objective function value. Therefore, given a partial solution under construction M and a candidate to be included on it c, the greedy function $g(c)$ which evaluates the relevance of inserting c in M is evaluated as follows:

$$g(c) = \sum_{i \in M} d_{ci}$$

It is worth mentioning that this greedy function allows us to perform an efficient evaluation of the objective function. In particular, after inserting the next candidate c in the solution under construction, it is not necessary to evaluate the complete objective function from scratch. Instead, the objective function value of the solution under construction M is updated as $MDP(M) + g(c)$. This optimization in the objective function evaluation reduces its complexity from $O(n^2)$ to $O(n)$, thus reducing the computing time required to construct a solution.

In order to define a local search for the MDP, it is necessary to describe the main elements that will conform it: the move operator, the neighborhood explored, and the strategy considered to explore the neighborhood. The move operator selected for this work is the interchange move, which consists of replacing an element already included in the solution with another element which has not been selected yet. This move operator, named $Interchange(M, i, j)$, with $i \in M$ and $j \in V \setminus M$, has been shown to be effective in the context of diversity problems. Another advantage of this move operator is that it always produces a feasible solution, since the number of selected elements always remains the same.

Since the local search is the most computationally demanding stage in the GRASP methodology, it is recommended to optimize this process to reduce the computing time required by the complete GRASP algorithm. In a direct implementation, performing an interchange move will result in the evaluation of the new solution from scratch. However, in the context of MDP, it is possible to leverage a factorization in the evaluation of the objective function that allows us to avoid evaluating the complete solution from scratch. In particular, it is only necessary to evaluate the distance between the node being removed and the nodes remaining in the solution and the distance between the node being added and the nodes remaining in the solution. Then, the evaluation of the solution M', resulting from performing $Interchange(M, i, j)$ is evaluated as follows:

$$MDP(M') \leftarrow MDP(M) + \sum_{k \in M} d_{jk} - d_{ik}$$

This optimization considerably reduces the computational effort required to execute the complete local search, thus reducing the computing time of the complete GRASP procedure.

Having defined the move operator, the next step consists of defining the neighborhood that will be explored with that operator. In this work, the local search explores the neighborhood N_i, generated by all the solutions that can be reached with a single interchange move. More formally,

$$N_i(M) \leftarrow \left\{ M' \leftarrow Interchange(M, i, j) \quad \forall i \in M \wedge \forall j \in V \setminus M \right\}$$

Finally, it is necessary to propose a strategy to traverse the proposed neighborhood. As stated in Section 5.1, two main strategies are usually considered when designing local search methods. This work considers both strategies for solving the MDP in order to provide a comparison between them in terms of both quality and computing time.

5.4 Computational Experiments

The main objective of this section is to evaluate the performance of GRASP metaheuristic when dealing with the MDP. The set of instances considered for this experimentation has been derived from the MDPLIB 2.0, which is publicly available.[1] In particular, we have considered a subset derived from sets GKD-c, MDG-a, and MDG-b, conforming a dataset of 68 instances. The experiments are divided into two different sections: preliminary and final. The former are devoted to tune the parameters of the GRASP algorithm, while the latter is designed to evaluate the performance of the best configuration of GRASP. In order to avoid overfitting, the preliminary experiments are performed over a subset of 20 representative instances.

[1] https://www.uv.es/rmarti/paper/mdp.html.

All the experiments report the following metrics: *Diversity*, the average objective function value among all the instances considered in the experiment; *Time (s)*, the average computing time required by an algorithm measured in seconds; *Dev. (%)*, the average deviation with respect to the best solution found in the experiment, evaluated as $(Best - Obj.Func.)/Best$, where *Obj.Func.* refers to the objective function value obtained by the algorithm; and #Best, the number of times that the algorithm reaches the best solution of the experiment.

Since each iteration of GRASP, consisting in a construction and a local improvement, is totally independent from the other iterations, we have decided to execute it in parallel, considering that, nowadays, every computer have more than one CPU available which can be used to accelerate the experiments. All the algorithms have been coded in Java 17 and the experiments have been executed in an AMD EPYC 7282 with 16 GB RAM and 16 processors. Therefore, it is able to perform 16 simultaneous iterations of GRASP. Synchronization procedures have been considered to recover the best solution found among all iterations. The code and individual results for each instance are publicly available.[2]

5.4.1 Preliminary Experiments

The preliminary experiments are designed with the aim of tuning the parameters of GRASP to produce better solutions. In the context of the algorithm proposed in Section 5.3, there are two decisions to be made when executing GRASP: the value of the α parameter and the local search strategy selected (first or best improvement). As it is customary in GRASP, each algorithm generates 100 solutions.

The first experiment is devoted to select the most adequate value for the α parameter. In particular, the tested values are $\alpha = \{0.25, 0.50, 0.75, RND\}$, where *RND* indicates that the value is selected at random for each construction in the range [0,1] following a uniform distribution. The adjustment of this parameter is usually done by executing the constructive procedure in isolation, without coupling it with the local search method. However, this is not a good practice, since the main objective of the constructive phase of GRASP is not only to generate high-quality solutions but also diverse ones. Therefore, selecting the value of α providing the largest quality may result in a constructive procedure without diversification. As a consequence, it is recommended to evaluate the appropriateness of an α-value also considering the local search phase. To illustrate this behavior, both experiments are performed in this work. First of all, Table 5.1 shows the results obtained when executing the constructive procedures without considering any local search method.

As it can be derived from the table, the best value for α is *RND*. It is an expected result, since that value explores the complete range of available values for α. If we

[2] https://grafo.etsii.urjc.es/G4D.

Table 5.1 Results of the constructive procedures executed in isolation. Best results are highlighted in bold font

Alpha	Diversity	Time (s)	Dev(%)	#Best
RND	**52,638.32**	**0.01**	**0.00**	**19**
0.25	51,711.41	0.01	2.16	1
0.50	47,065.91	0.01	11.38	0
0.75	44,202.31	0.01	18.34	0

Table 5.2 Results of GRASP considering the first improvement approach in the local search method. Best results are highlighted in bold font

Alpha	Diversity	Time (s)	Dev(%)	#Best
RND	53,603.41	0.76	0.05	13
0.25	53,488.4	0.59	0.06	13
0.5	**53,620.17**	**0.91**	**0.04**	**16**
0.75	53,584.44	0.98	0.20	13

Table 5.3 Results of GRASP considering the best improvement approach in the local search method. Best results are highlighted in bold font

Alpha	Diversity	Time (s)	Dev (%)	#Best
RND	**53,698.7**	**0.92**	**0.05**	**18**
0.25	53,359.57	0.62	0.10	16
0.5	53,536.3	1.12	0.07	17
0.75	53,514.15	1.25	0.06	17

do not consider that value, the smaller the value of α, the better, indicating that greedy values provide better results, as expected.

The next experiment is devoted to evaluate the performance of the constructive procedures when coupling them with the local search method, considering the first improvement approach. Table 5.2 shows the results obtained when considering the first improvement approach in the local search phase of GRASP.

It is worth mentioning that the improvement phase is able to balance the quality obtained by different α-values. This results are in line with the idea of not necessarily selecting the best constructive procedure in terms of quality without considering diversity. In particular, $\alpha = 0.5$ is now the best value for the α-parameter, demonstrating how increasing diversity may eventually lead to better solutions. Notice that the local search is able to considerably improve the results of the constructive procedure in all the α-values, resulting in rather similar results.

The next experiment is devoted to analyze the performance of the best improvement approach inside the GRASP framework. Table 5.3 shows the results obtained when considering this local search method.

In this case, the best results are obtained when considering $\alpha = RND$, being able to reach the best objective function value in 18 out of 20 instances, and obtaining a deviation of 0.05, which indicates that, in those instances in which it is not able to reach the best solution, it still remains really close to it. Without considering $\alpha = RND$, the quality of the results increases with the value of α, indicating that diversification is a key part of the algorithm.

In both, first improvement and best improvement experiments, although all the variants require similar computing times, it can be seen how increasing the randomness (i.e., increasing the value of α) leads to more time-consuming approaches. This behavior can be partially explained by the situation of the starting point from which the local search starts in each case. Values close to 1 indicate that the initial solutions will be diverse but not necessarily of high quality, so the local search will need more iterations to reach a local optimum.

5.4.2 Final Experiment

Having selected the best value for the α-parameter in both, first and best improvement strategies, the objective of this final experiment is to perform a competitive testing between the considered strategies. In this case, the complete set of instances is considered.

The comparison between first and best improvement is still an open issue which should be experimentally solved for each problem, since it is hard to provide conclusive hypothesis without performing an experimental comparison. To that end, Table 5.4 shows the results obtained by GRASP when considering first and best improvement strategies in the local improvement phase over the complete set of instances.

Analyzing the results, it can be concluded that both algorithms present similar performance in terms of quality and computing time. Although it is expected that best improvement require larger computing times than first improvement, it usually gets trapped in local optima faster. Therefore, first improvement usually is able to perform more iterations, reaching equivalent computing times. If we analyze the quality of each algorithm, first improvement is able to reach a larger number of best solutions (56 versus 43) and a slightly smaller deviation (0.05% versus 0.10%). The small deviation value in both algorithms indicates that in those instances in which they are not able to reach the best solution, they still present competitive results.

With the aim of validating these results, we have performed the nonparametric pairwise Wilcoxon signed rank test. The resulting p-value, smaller than 0.05, indicates that there are statistically significant differences between both algorithms. Additionally, results presented in Table 5.5 show how the first improvement variant is able to reach strictly better solutions than best improvement in 44 out of 68 instances, while the best improvement approach only reaches 19 strictly better solutions than first improvement, with 5 ties.

Table 5.4 Comparison of GRASP considering best and first improvement strategies with the best value of α for each variant

Algorithm	Diversity	Time (s)	Dev (%)	#Best
$GRASP_{BI}$	124,124.89	1.56	0.10	43
$GRASP_{FI}$	124,158.68	1.52	0.05	56

Table 5.5 Results of Wilcoxon signed rank test comparing $GRASP_{FI}$ with $GRASP_{BI}$

	N	Mean rank	Sum of ranks
$GRASP_{FI} < GRASP_{BI}$	19	34.95	664.00
$GRASP_{FI} > GRASP_{BI}$	44	30.73	1352.00
Ties	5		
Total	68		

These results indicate that the first improvement approach is able to produce better results in terms of quality when comparing it with the best improvement variant, both requiring similar computing time.

5.5 Conclusions

This work has presented a detailed review on GRASP for the diversity problem and a particular design of a GRASP procedure for solving the maximum diversity problem. The proposal conforms a greedy randomized constructive procedure , whose greedy function coincides with the objective function and two local improvement methods: first and best improvement. The detailed experimental comparison shows the superiority of the first improvement approach, analyzing the contribution of each part of the algorithm. This design is not oriented to be competitive with the state of the art, but to show the simplicity of GRASP, which is a simple yet effective metaheuristic for solving a large variety of hard combinatorial optimization problems.

Acknowledgments This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación (MCIN/AEI/10.13039/501100011033/FEDER, UE) under grant ref. PID2021-126605NB-I00 and PID2021-125709OA-C22, Comunidad de Madrid y Fondos Estructurales de la Unión Europea with grant ref. P2018/TCS-4566, and Generalitat Valenciana with grant ref. CIAICO/2021/224.

References

Benoist, T., Estellon, B., Gardi, F., Megel, R., Nouioua, K.: Localsolver 1. x: a black-box local-search solver for 0-1 programming. 4or **9**(3), 299–316 (2011)

Duarte, A., Martí, R.: Tabu search and grasp for the maximum diversity problem. Eur. J. Oper. Res. **178**(1), 71–84 (2007)

Duarte, A., Sánchez-Oro, J., Resende, M.G., Glover, F., Martí, R.: Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization. Informat. Sci. **296**, 46–60 (2015)

Feo, T.A., Resende, M.G.: A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. **8**(2), 67–71 (1989)

Feo, T.A., Resende, M.G., Smith, S.H.: A greedy randomized adaptive search procedure for maximum independent set. Oper. Res. **42**(5), 860–878 (1994)

Festa, P., Resende, M.G.: An annotated bibliography of grasp-part i: algorithms. Int. Trans. Oper. Res. **16**(1), 1–24 (2009a)
Festa, P., Resende, M.G.: An annotated bibliography of grasp-part ii: applications. Int. Trans. Oper. Res. **16**(2), 131–172 (2009b)
Ghosh, J.B.: Computational aspects of the maximum diversity problem. Oper. Res. Lett. **19**(4), 175–181 (1996)
Hansen, P., Mladenović, N.: First vs. best improvement: an empirical study. Discrete Appl. Math. **154**(5), 802–817 (2006)
Hart, J.P., Shogan, A.W.: Semi-greedy heuristics: an empirical study. Oper. Res. Lett. **6**(3), 107–114 (1987)
Martí, R., Sandoya, F.: Grasp and path relinking for the equitable dispersion problem. Comput. Oper. Res. **40**(12), 3091–3099 (2013)
Martí, R., Gallego, M., Duarte, A., Pardo, E.G.: Heuristics and metaheuristics for the maximum diversity problem. J. Heurist. **19**(4), 591–615 (2013)
Martí, R., Martínez-Gavara, A., Sánchez-Oro, J.: The capacitated dispersion problem: an optimization model and a memetic algorithm. Memet. Comput. **13**(1), 131–146 (2021)
Martínez-Gavara, A., Campos, V., Laguna, M., Martí, R.: Heuristic solution approaches for the maximum minsum dispersion problem. J. Global Optim. **67**(3), 671–686 (2017)
Peiró, J., Jiménez, I., Laguardia, J., Martí, R.: Heuristics for the capacitated dispersion problem. Int. Trans. Oper. Res. **28**(1), 119–141 (2021)
Resende, M.G., Martí, R., Gallego, M., Duarte, A.: Grasp and path relinking for the max-min diversity problem. Comput. Oper. Res. **37**(3), 498–508 (2010)
Santos, L., Ribeiro, M.H., Plastino, A., Martins, S.L.: A hybrid grasp with data mining for the maximum diversity problem. In: International Workshop on Hybrid Metaheuristics. Springer, Berlin, pp. 116–127 (2005)
Silva, G.C., Ochi, L.S., Martins, S.L.: Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In: International Workshop on Experimental and Efficient Algorithms. Springer, Berlin, pp. 498–512 (2004)
Silva, G.C., De Andrade, M.R., Ochi, L.S., Martins, S.L., Plastino, A.: New heuristics for the maximum diversity problem. J. Heurist. **13**(4), 315–336 (2007)