Optimizing GRASP with Path Relinking Strategies for the Maximum Diversity Problem

Josep Peiró Ramos & Olaf Meneses Albalá

April 2025

Contents

1 Definition of the problem

The problem we address is called the *Maximum Diversity Problem (MDP)*. It involves selecting a subset of m items from a larger set of n items, where $m \ll n$, in a way that maximizes the overall diversity among the selected items. Diversity between two items is defined as the distance between them, typically using a metric such as the Euclidean distance. A higher total pairwise distance among selected items indicates greater diversity. Therefore, the objective of MDP is to choose a subset of items that maximizes the sum of distances between all pairs in the subset.

The MDP has practical applications in areas such as plant breeding, social studies, ecological preservation, product design, workforce management, and curriculum design.[2]

In our work, we used 15 problem instances: 6 of which had $n = 100$ items and a subset size of $m = 10$; and 9 instances with $n = 500$ items and a subset size of $m = 50$.

2 State of the art

The *Maximum Diversity Problem (MDP)* has been extensively discussed since it can be easily formulated [2]. There have been many heuristic proposals for solving it[4], from iterated greedy methods[8], Tabu Search [1],[3], or evolutionary algorithms[6]. Also, it can be solved by an exact method based on branch and bound[9].

In these experiments, we limit ourselves to comparing two heuristic methods in terms of reaching the best objective value, given a fixed computational time. In particular, we want to determine which method is better: a full-GRASP method or a mixed algorithm that creates many constructive solutions with GRASP and link some of them by a Path Relinking method [5], an approach previously investigated in the literature [7].

The first step during our experiments was to find the best GRASP algorithm for the instances described in the above section. For that, we used a simple GRASP algorithm, which creates a first partially random solution and improves it via a greedy local search, removing the worst element from the current solution and adding the best candidate from the nonselected items.

To do so, it was necessary to calibrate the parameter α, which controls the degree of randomness in the construction phase. After that, we could start comparing it against the implementation with Path Relinking.

Additionally, our investigation was not restricted to a single Path Relinking method; we experimented with several alternative strategies.

3 Calibrating α

The methodology used starts with a GRASP constructive. It starts by selecting a random element from the initial set. Then, as is common in GRASP algorithms, instead of selecting the next item as the one with the longest distance, we create a list of candidates, which contains a portion of the farthest elements and randomly select one of them. The size of this portion is determined by the parameter α. If $\alpha = 0$, the constructive is deterministic, and if $\alpha = 1$, it is random.

We first determined the best value of these parameters. To do so, we ran the GRASP algorithm with a greedy LocalSearch with many different values of α for every instance. We did this, allowing the algorithm to work for different quantities of time, from 0.1 seconds to 1 minute per instance, so we can also compare the performance with different running times.

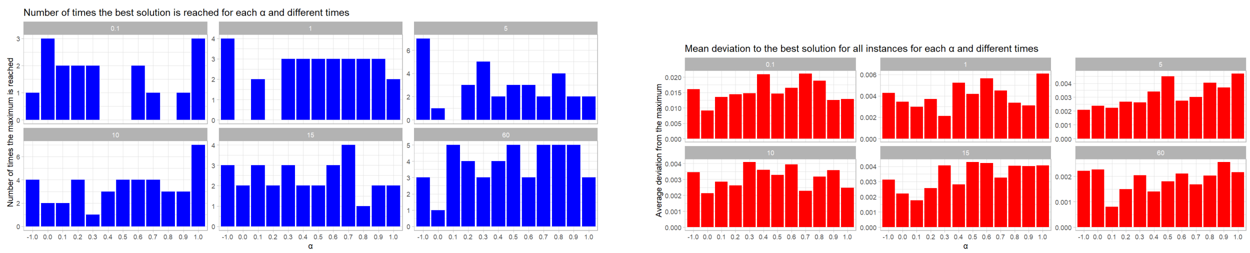
To figure out which setting (α) works best for our algorithm, we looked at two metrics:

1. **How often it wins**: We counted the number of times a specific setting produced the absolute best result for a particular problem, compared to all the other settings we tried for that same problem.

2. **How close it gets (on average)**: For each problem, we first identify the best result achieved by any setting. Then, for a specific setting, we measured how far its result was from that best result. We calculated the average of these differences across all problems.

In short, we want a setting that wins often (Metric 1) and, even when it doesn't win, doesn't fall too far behind the winner (Metric 2).

Even though the results are not clear, as the values are distributed very uniformly, we identified $\alpha = 0.1$ as the best value, especially when the maximum time is increased. All results are visible in Figure 1.



(a) Number of times the best solutions is reached using α.

(b) Average deviation from the best solution using different values of α.

Figure 1: Graphs obtained comparing metric for finding the best value of α for the GRASP construction. We run the algorithm with 15 instances varying the maximum allowed computing time. On the right, the number of times the best value was obtained using the value of α, which we want to maximize. On the left, the deviation from the best, which we try to minimize. On top of each subplot, the maximum time allowed in seconds is indicated. Value -1 means random value of α.

.

4 Path Relinking

Path Relinking (PR) was integrated into our GRASP algorithm as an intensification strategy, serving as an alternative or complement to a standard greedy Local Search phase. The goal of PR is to explore the solution space between two known high-quality but distinct solutions.

4.1 First idea: Basic Path Relinking

In our first implementation, each GRASP iteration begins by constructing a diverse set of candidate solutions using the randomized greedy constructive phase. From this set, we identify two key solutions:

1. **The Elite Solution:** The best solution found in the set based on the objective function.
2. **The Guiding Solution:** The solution within the set that is most different from the Elite Solution (details on distance measurement in Section 4.1.1).

Path Relinking then generates a trajectory of solutions starting from the Guiding Solution and moving towards the Elite Solution. This is achieved by iteratively introducing elements present in the Elite Solution into the current solution while removing elements not present in the Elite Solution. Each intermediate solution generated along this path is evaluated. The best solution encountered during this process, potentially better than either the initial Elite or Guiding solutions, is saved.

This method assumes that even better solutions might exist along the path connecting two existing good solutions. Although improvement is not guaranteed, PR provides a structured way to investigate promising regions of the solution space between diverse, high-quality starting points. The best solution found via PR can then update the overall best solution found so far before the next GRASP iteration begins.

4.1.1 Measuring distances between solutions

To find the "most different" or "farthest" solution from the elite one, we need a measure of distance or dissimilarity. Since our solutions are represented as sets of elements (identified by numbers), we measure the size of the intersection between the two sets: a smaller intersection size implies greater dissimilarity (higher distance). Therefore, the Guiding Solution selected is the one whose element set has the smallest intersection with the Elite Solution's element set.

4.2 Improving constructive: Local Search before PR

The next modality we added to the simplest algorithm was the possibility to carry out a Local Search for every GRASP solution from the initial set of solutions before the Path Relinking. We assume that this will consume computational time but will guarantee an improvement in the solutions for Path Relinking. As we ensure the Linking between the best solution and the most distant one, we want to check the hypothesis that a better solution may be located between both of them.

4.3 Leaving time for the constructive

Until now, we allowed the GRASP constructive only to create a certain and fixed number of solutions per iteration. A way of making more flexible constructions is by creating as much as possible in a quantity of time, which becomes determined as a proportion of the total time, i.e. if we specify 10 seconds for the algorithm to run, and 10% of the time for the constructive, the GRASP will create as many initial solutions as possible in 1 second, then it will carry out the Path Relinking and the process will start again until the total amount of time ends. This time also involves the Local Search explained in the previous section.

4.4 Refining the path: Local Search with PR

Another complementary approach consists of applying Local Search during the Path Relinking. As the algorithm generates intermediate solutions along the connecting path, it evaluates them; if an intermediate solution is better than the best solution on the path so far, a greedy Local Search is immediately applied starting from that point.

Using a mountain analogy, the path between two good solutions might cross a mountainside instead of the peak. Applying Local Search from this path is like climbing uphill from the mountainside, seeking a higher peak (a better solution).

4.5 Randomizing paths: PR with random candidates

Another factor we decided to add, which is crucial for the correct performance of Path Relinking is, besides creating the path between the best solution and the most distant one, also selecting a set of solutions from the initial family constructed by the GRASP method and a corresponding pair, which is called the elite set.

In order to create the subset of candidates for carrying out Path Relinking, we choose randomly (without replacement). The corresponding pair of each candidate will be the most distant solution from the whole set. This alternative approach will ensure a less greedy behavior, since it is allowing to search different solutions from the current best one.

We introduce the elite set size as a new parameter for this method. Keep in mind that the size of the elite set cannot exceed the number of solutions initially created.

5 Experiment

As we explained in the first section, we have 15 different instances with sizes of 100 and 500 items to select and a selection set size of 10 and 50, respectively.

Combining all the variants explained in the previous section, we have the following parameters:

- **GRASP Constructive:** We tested four strategies for generating the initial solution pool:
  - Fixed number of solutions: 10 or 20.
  - Proportion of total execution time allocated: 0.05 (5%) or 0.1 (10%). We limited this to a maximum of 10% to reserve sufficient time for subsequent phases.
- **Pre-PR Local Search:** A boolean parameter (True/False) indicating whether Local Search is applied to each initial GRASP solution *before* Path Relinking.
- **Intra-PR Local Search:** A boolean parameter (True/False) indicating whether Local Search is applied to improving solutions found *during* the Path Relinking process.
- **Elite Set Size (Path Relinking):** The number of elite solutions used to diversify the Path Relinking search, beyond the initial and guiding solutions. Tested values: 0 (representing Greedy Path Relinking), 2, 3, 4, and 5.
- **Execution Time Limit:** Maximum runtime allowed per trial. Tested values: 0.1, 1, 5, 10, 15, and 60 seconds.

Combining these parameter variations yields $4 \times 2 \times 2 \times 5 \times 6 = 480$ unique configurations per instance. Executing each configuration across all 15 instances resulted in a total of $15 \times 480 = 7200$ observations.

6 Prior experimentation

6.1 Determining significance of Local Search

Once we got every observation, we determined the utility of applying Local Search both before and after PR. To do so, we conducted a series of frequentist hypothesis tests.

6.1.1 Test for LS before PR

First, we noted whether applying Local Search before (and only before) Path Relinking improved the algorithm's performance. So we took the observations where we did not apply LS after and compared, maintaining the rest of parameters equal, the solution with and without LS.

For every pair of solutions with the same parameters with and without LS, we calculated the difference of the target value obtained ($Solution\ with\ Local\ Seach\ -\ Solution\ without\ Local\ Seach$). If the difference is above 0 means that the solutions obtained with Local Search are higher, so better, than the ones without. similarly, if it is 0 or below would mean that there is no significant difference, so it is not useful.

We assumed the differences to be normal. Then checked by a t-test if they are greater than 0 with a significance over 95% (i.e. $\gamma = 0.05$). We obtained a p-value extremely low: $4.47 \cdot 10^{-234}$. Therefore, we can safely assume that applying Local Search before Path Relinking in order to improve the GRASP solutions improves the performance.

6.1.2 Test for LS after PR

Similarly to our previous test, we also tested if applying Local Search after Path Relinking improves solutions.

We compared solutions with and without post-construction Local Search, focusing only on runs where Local Search was initially skipped during construction, and calculated the difference in their target values. We did the same test as the one explained above and got a p-value of $6.3 \cdot 10^{-163}$. Then, applying Local Search during the Path Relinking process also improves performance.

6.1.3 Test for combined LS

We also investigated applying Local Search (LS) simultaneously before and after the main process. Hypothesis tests compared this "both stages" approach against applying LS only before ($p = 0.026$) and only after $p = 1.91 \times 10^{-36}$. Both comparisons showed statistically significant differences, indicating that applying LS in both stages yields superior results (details in Table 1). Consequently, we enabled LS application in both stages.

Parameters	p-value	Avg. difference	Prop. higher values
$\overline{Bef},\ \overline{Aft}\ vs\ Bef,\ \overline{Aft}$	$4.476 \cdot 10^{-234}$	33.841	0.821
$\overline{Bef},\ \overline{Aft}\ vs\ \overline{Bef},\ Aft$	$6.304 \cdot 10^{-163}$	25.666	0.699
$Bef,\ \overline{Aft}\ vs\ Bef,\ Aft$	0.026	1.198	0.419
$\overline{Bef},\ Aft\ vs\ Bef,\ Aft$	$1.913 \cdot 10^{-36}$	9.373	0.628

Table 1: Comparison of solution quality with and without Local Search (LS). The table analyzes the difference in objective values when LS is applied versus not applied during the constructive phase (Bef vs. $\overline{Bef}$) and Path Relinking (Aft vs. $\overline{Aft}$). It shows the mean difference, the proportion of instances where the difference is positive, and the p-value for the hypothesis that the mean difference is significantly greater than zero.

6.2 Best elite set size

We needed to find the best settings for our elite set, which involves two choices:

1. **How many initial solutions to generate:** We tested using fixed numbers (10 or 20) and proportions of the total time (5% or 10%).
2. **How many of the best solutions to keep for Path Relinking:** We tested keeping 0, 2, 3, 4, or 5 solutions.

Since we tested these combinations across 6 different time limits (0.1s to 60s), we had many results. To choose the best parameter combination for each time limit, we did the following, similar to the previous methodology for finding the best α:

1. **Find the "Best Known" Solution:** For each problem instance and time limit, we identified the highest objective value achieved by *any* parameter combination.
2. **Evaluate each combination:** We assessed each parameter combination based on two metrics that have already appeared in this document:
   - **Frequency:** How often did it find the "Best Known" solution? (Higher is better)
   - **Deviation:** What was its average difference from the "Best Known" solution across all instances? (Lower is better)
3. **Select Top Performers:** For each time limit, we listed the top 5 combinations for maximizing frequency and the top 5 for minimizing deviation.
4. **Final Choice:** The "best" parameter combinations were those that appeared in *both* top 5 lists.

Essentially, we looked for parameters that consistently found the best solutions and stayed close to the best solutions when they didn't find the absolute best.

$$\text{Best parameters} = \{\text{Top 5 by Frequency}\} \cap \{\text{Top 5 by Deviation}\}$$

For each running time, we found a subset with the two best combinations for each running time, as shown in Table 2. From the table results, we observe that it is better to set a fixed number of constructions than setting it as a function of time. Only one out of the twelve best combinations was built with the parameter of 5% of the maximum time.

Moreover, as it is observed from the combinations obtained, when the running time is lower, it is better to have a bigger constructive, as most of the combinations for 0.1, 1, and 5 seconds prefer to create 20 solutions each iteration. In contrast, as the allowed execution time increases, the value of 10 solutions per iteration appears more often.

However, the results obtained show that a large elite set size (4-5) is more common between the best combinations, especially when more executing time is permitted. This demonstrates the importance of exploring and avoiding greed.

In summary, optimal parameters depend on the runtime. Short runs (0.1, 1, 5 seconds) favor generating many constructive solutions (e.g., 20) due to limited exploration time. Longer runs allow for more thorough exploration, favoring fewer constructions per iteration (e.g., 10) but emphasizing a large elite set (e.g., 4-5) to maintain solution quality and diversity.

Running time	Constructive Size	Elite Set Size	Freq Highest value	Avg. Deviation from best
0.1	Nº sols: 20	3	2	0.00418
	Nº sols: 20	0	2	0.00415
1	Nº sols: 20	4	3	0.00257
	Nº sols: 10	5	2	0.00246
5	Nº sols: 20	3	4	0.00318
	Prop: 0.05	3	2	0.00393
10	Nº sols: 10	5	5	0.00264
	Nº sols: 10	4	3	0.00262
15	Nº sols: 10	4	6	0.00236
	Nº sols: 20	4	4	0.00240
60	Nº sols: 10	4	7	0.00158
	Nº sols: 20	5	5	0.00148

Table 2: Parameter settings excelling in both solution frequency and low deviation across different time limits.

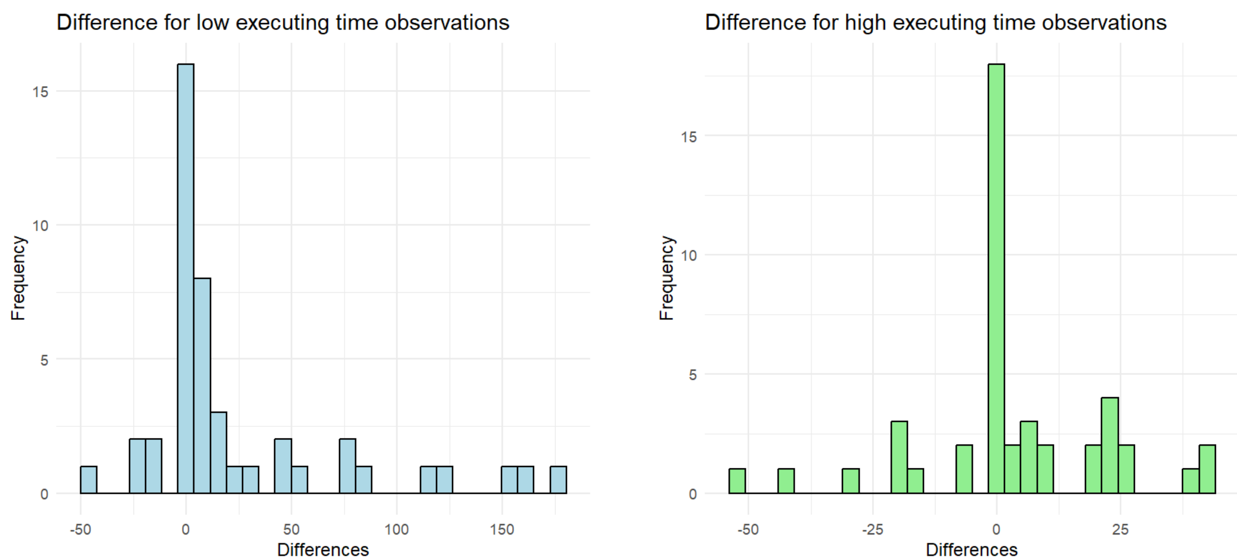
t

7 Results

After exploring the best combination of parameters, we tested whether applying Path Relinking improves performance. To evaluate the impact of Path Relinking, we conducted another t-test, similar to the one described in Section 6.1. This time, we compared the performance of GRASP using its best-identified parameter ($\alpha = 0.1$) against the results obtained when combining GRASP with Path Relinking.

Since there is no single best parameter combination for Path Relinking across all running times, we selected parameters based on the results in Table 2. For shorter running times (0.1, 1, and 5 seconds), we used a constructive size of 20 and an elite set size of 3, as these were common values for these durations. For longer running times (10, 15, and 60 seconds), we opted for a constructive size of 10 and an elite set size of 4. In all Path Relinking experiments, Local Search was applied both before and after the Path Relinking step.

After selecting the parameters and obtaining the corresponding results for each instance, we calculated the difference between the objective value achieved with Path Relinking and the value achieved with GRASP alone. We then applied a t-test to determine whether this difference is statistically significantly greater than zero. It is important to note that the number of samples for this comparison is somewhat limited (45 observations per histogram, resulting from 3 running times across 15 instances). Despite this limitation, the distributions of these differences are visualized in Figure 2.



(a) Distribution of the difference when the execution is short (0.1, 1 and 5 seconds).

(b) Distribution of the difference when the execution is long (10, 15 and 60 seconds).

Figure 2: Distribution of the differences between the objective value using Path Relinking minus the objective value achieved only via GRASP.

For the first case, analyzing the results from short running times, the calculated p-value is 0.00073203. Since this is less than the significance level ($\gamma = 0.05$), we reject the null hypothesis. Notably, the mean difference between the Path Relinking results and the GRASP-only results is 25.321. This indicates that, on average, incorporating Path Relinking significantly improved the objective function value by approximately 25.3 points for these shorter durations.

In the second case, the p-value is $0.132301 > \gamma = 0.05$. However, if we take the results of the times limits of 10 and 15 seconds, with the same parameters, the p-value turns to $0.45 < 0.05$. This suggests that as execution times become longer, particularly for these relatively small instances, the performance difference between the two approaches may decrease. With more time, both methods have a greater chance of converging towards high-quality or optimal solutions, potentially reducing the relative advantage gained by Path Relinking.

Even with this trend towards convergence, the data from the three longest execution periods still **shows a benefit**: Path Relinking achieved an average improvement of **3.195 points** and obtained **higher objective values in 77%** of the instances.

8 Conclusion

This study investigated the effectiveness of integrating Path Relinking (PR) into a GRASP framework for solving the Maximum Diversity Problem (MDP), comparing it against a standalone GRASP approach. Our experiments began by calibrating the GRASP constructive phase, identifying $\alpha = 0.1$ as the most effective parameter value.

We explored several variations of the GRASP+PR algorithm, focusing on the application of Local Search (LS) and the configuration of the constructive and PR phases. Statistical analysis confirmed that applying LS both before PR (to refine initial solutions) and during PR (as an intensification step on improving path solutions) significantly improves performance compared to applying LS at only one stage or not at all.

Moreover, we also observed that the optimal configuration for the constructive phase (number of initial solutions) and the PR elite set size is dependent on the allocated runtime. Shorter execution times benefit from generating a larger pool of initial solutions (e.g., 20) with a moderately sized elite set (e.g., 3), while longer runtimes favor fewer initial constructions (e.g., 10) combined with a larger elite set (e.g., 4-5) to promote diverse exploration. Using a fixed number of constructions generally outperformed allocating a proportion of time.

Comparing the optimized GRASP+PR (with dual LS and runtime-adapted parameters) against the best standalone GRASP ($\alpha = 0.1$), we found that the hybrid approach provides statistically significant improvements, particularly under shorter time constraints. Although the performance gap decreases with longer execution times as both methods approach convergence, GRASP+PR consistently demonstrated an advantage, achieving higher objective values on average and in the majority of instances even at the longest tested duration.

In conclusion, Path Relinking, when improved with strategically applied Local Search and tuned according to available runtime, proves to be a powerful addition to GRASP for solving the Maximum Diversity Problem.

References

[1] R. Aringhieri, R. Cordone, and Y. Melzani. Tabu search vs. grasp for the maximum diversity problem. *4OR: A Quarterly Journal of Operations Research*, 6(1):45–60, 2008.

[2] F. Glover C.C. Kuo and K. S. Dhir. Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185, 1993.

[3] A. Duarte and R. Martí. Tabu search and grasp for the maximum diversity problem. *European Journal of Operational Research*, 178:71–84, 2007.

[4] F. Glover, Ching-Chung Kuo, and K. Dhir. Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences*, 1998.

[5] Fred Glover, Manuel Laguna, and Rafael Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):653–684, 2000.

[6] K. Katayama and H. Narihisa. An evolutionary approach for the maximum diversity problem. In W. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent advances in memetic algorithms*, volume 166, pages 31–47. Springer, Berlin, 2005.

[7] M Gallego M. G. C. Resende, R. Martí and A Duarte. Grasp and path relinking for the max–min diversity problem. *Computers & Operations Research*, 37(3):498–508, Mar 2010.

[8] C. García-Martinez M. Lonzano, D. Molina. Iterated greedy for the maximum diversity problem. *European Journal of Operational Research*, 214(2011):31–38, 2011.

[9] A Duarte R Martí, M Gallego. A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research*, 200(1):36–44, 2010.