

There is a homework (project) assignment and will count towards the final grade (50%). It basically consists on implementing a heuristic algorithm to solve a combinatorial optimization problem. Homework assignments will be submitted via e-mail to rafael.marti@uv.es.

We developed in the class a Visual Basic code that implements a GRASP for the Maximum Diversity Problem (MDP). The students can use this implementation as an example, or as a part of their own implementation based on another metaheuristic technology for the MDP. Homework will be prepared individually or in groups of 2 or 3 students, and **submitted by August 6th**.

Students have to choose between the two options below. They have two weeks (after the termination of the course) to complete it, and send two files (code and report). The code can be in Visual Basic for Excel or any other programming language. The report is a simplified version of a typical journal paper on heuristic optimization. It will have between 5 and 10 pages, describing the problem, the proposed method and an empirical comparison of this method with the GRASP developed in the class over the 5 instances provided with $n=500$ and $m = 40$.

OPTION 1 – PATH RELINKING

We developed in the class a Visual Basic code that implements a GRASP for the Maximum Diversity Problem. This code solved an instance with 100 elements. You have to adapt it to solve the 5 instances of size 500 that you can find in the course folder. You have to test different values of the *alpha* parameter and report the associated results.

You have to develop a method to create paths between elite solutions collected with the application of the GRASP. Paths are obtained in a similar way than standard local search but with a specific target (guiding solution), as determined by the Path Relinking methodology.

OPTION 2 –TABU SEARCH

You have to create a code that implements a TABU SEARCH for the Maximum Diversity Problem. You can use the pseudocode outline below for your own implementation. This code has to solve the 5 instances of size 500 that you can find in the course folder. You have to test different values of the *tabu tenure* parameter and report the associated results.

-
- Generate a random initial solution.
 - For n_iter iterations do
 - Explore the neighborhood of the current solution.
 - If there is an improving non-tabu move, perform it.
 - Otherwise, perform the best non-tabu move, even if it deteriorates the solution.
 - Add to the tabu list the element removed from the solution.
 - Update the current solution and its value.
 - If the current solution improves the best-found so far, update the best-found.
 - Return the best-found solution
-