Comparing Variable Width Backtracking and Metaheuristics, Experiments with the Maximum Diversity Problem

José-Matías Cutillas-Lozano, Miguel-Ángel Franco, Domingo Giménez

Department of Informatics and Systems, University of Murcia, Spain

Supported by Spanish MINECO and European Commission FEDER funds, TIN2012-38341-C04-03

Motivation

Due to the high computational cost of exact methods, the solution of large NP-hard optimization problems has traditionally been approached with heuristic methods, which can provide satisfactory solutions in short execution times, but with no guarantee of evaluation of all the possible solutions.

On the other hand, some exact methods, like backtracking, analyze all the solutions and find the optimum one, but with unaffordable execution times. We compare the application of metaheuristic methods and a modification of Backtracking (VWB) to the Maximum Diversity Problem (MDP). VWB begins by analysing a part of the solutions tree and successively increases the search to the whole solutions space.

Variable Width Backtracking

Ideas for VWB:

- To obtain lower and upper bounds (LB and UB) of the value of the optimum solutions reachable for each node in the tree.
- To divide the work into successive steps, s, with each step consisting of a backtracking with the lower and upper bounds of each node in that step (LB_i and UB_i) for step i obtained in the interval of the complete bounds.
- The interval in a step includes the interval of the previous step, [LB_i, UB_i] ⊆ [LB_i+1, UB_i+1].
- The interval of the last step is equal to the complete interval, [LB_i, UB_i] = [LB, UB].

The tree is explored in successive steps, with possible improvements of the best solution in each step, and the optimum solution of the last step is that of the problem.

The search in each step is scattered throughout the tree in an attempt to avoid the temporal concentration of the search in certain areas of the tree, which could delay satisfactory solutions. The search in step i + 1 includes the nodes generated in previous steps, and the huge execution time of backtracking methods becomes even larger.

The only possible advantage of VWB is the early generation of satisfactory results.

For example:

- in a maximization problem the upper bound in iteration j could be LB + j * (UB - LB) / (s + 1).
- VWB consists of s backtracks, where each PartialBacktracking comprehends the code of the original backtracking without the initialization of the partial optimum solution and value (POV).

The generation of one node includes the updating of the upper bound of the node for this iteration and is pruned if its bound for this iteration is not higher than POV:

$$\text{LB}_{\text{node}} + j \cdot (\text{UB}_{\text{node}} - \text{LB}_{\text{node}}) / n \leq \text{POV}.$$