# Hyperheuristics based on parameterized metaheuristic schemes

José-Matías Cutillas-Lozano[1], Francisco Almeida[2], and Domingo Giménez[1]

[1] Departamento de Informática y Sistemas, Universidad de Murcia, Spain.
{josematias.cutillas,domingo}@um.es
[2] ETS Ingeniería Informática. Universidad de La Laguna, Tenerife, Spain.
falmeida@ull.es

## 1 Introduction

The use of a unified parameterized scheme for metaheuristics facilitates the development of metaheuristics and their application [1]. The scheme (Algorithm 1) considers a set of basic functions whose instantiation determines the particular metaheuristic that is being implemented.

**Algorithm 1** Parameterized metaheuristic scheme

```
Initialize(S,ParamIni)
while ( not EndCondition(S,ParamEnd)) do
  SS=Select(S,ParamSel)
  SS1=Combine(SS,ParamCom)
  SS2=Improve(SS1,ParamImp)
  S=Include(SS2,ParamInc)
end while
```

However, selecting the appropriate values of the *Metaheuristic parameters* ($ParamX$ in the algorithm) to apply a satisfactory metaheuristic to a particular problem can be difficult and is computationally demanding. The selection of these values can be made through a hyperheuristic method [2], which could in turn be developed with the same parameterized metaheuristic scheme. So the hyperheuristic works with the scheme at two levels: it is implemented on top of the parameterized metaheuristic scheme, selecting appropriate values for the metaheuristic parameters in the scheme, and consequently the metaheuristic itself.

In this work, different possibilities for application of hyperheuristics are considered. For example, there are different possibilities for the fitness function depending on whether the hyperheuristic is applied to one problem input in one execution (Fit1P1E), to several problem inputs in several executions (FitSPSE2), or to several problem inputs in one execution (FitSP1E). Furthermore, the hyperheuristic searches on a particular implementation of the metaheuristic scheme, and it has been tested with the application of local and global search methods (GRASP, Tabu Search, Genetic algorithms and Scatter Search), with a total of 20 metaheuristic parameters which allow us to experiment with different hybridations/combinations.

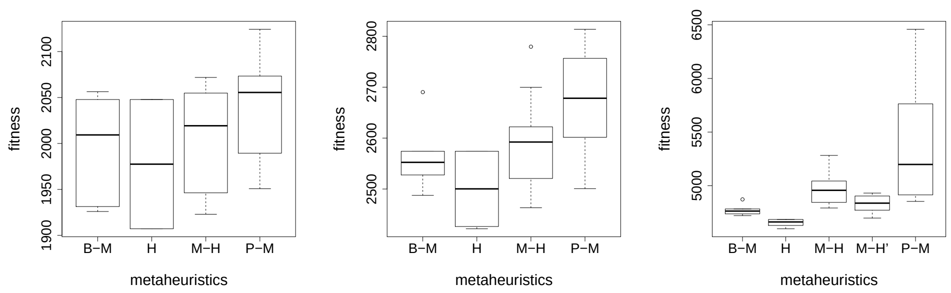
## 2 Experimental Results

For the experiments, we consider a problem of electricity consumption in exploitation of wells (PECEW) [3]. The water system consists of a series of pumps ($B$) of known power, located in wells, that draw water flow along a daily time range $R$. The total flow is the sum of the flows contributed by each well. The pumps may be running or idle at a given time. The pumps operate electrically and the electricity has a daily cost which should be minimized:

$$C_e = \sum_{i=1}^{R} \sum_{j=1}^{B} T_i P_j N_i X_{ij} \tag{1}$$

where $T_i$ is the cost of the electricity in the range $i$; $P_j$ is the electric power consumed by the pump $j$; $N_i$ is the number of hours of pump operation in the time slot $i$; and $X_{ij}$ represents a binary element of a matrix with values 1 or 0 for pump on or off. An element is represented by the binary matrix, $X$, of size $B \times R$. To study the application of hyperheuristics to the problem, six problem sizes $B - R$ are considered: 10-3, 10-6, 20-3, 20-6, 40-3 and 40-6. Experiments were carried out on a small NUMA system with 24 cores, with IntelXeon E7350 processors.

Figure 1 summarizes the results obtained when applying the hyperheuristics (H) or the metaheuristics obtained from the hyperheuristics (M-H), and compares them with those previously obtained with a direct application of pure metaheuristics (P-M) or with the best metaheuristic combinations (B-M). Experiments are carried out for several sizes of the problem PECEW, which are grouped into categories by number of pumps (for example, 20-x includes instances of size 20-3 and 20-6). Figures 1 (a) and (b) show the results when applying the M-H with fitness calculated as Fit1P1E and FitSPSE2, and Figure 1 (c) includes one additional set (M-H') which computes fitness as FitSP1E. M-H and M-H' are compared in (c) because there is a greater difference between the means of the groups studied for the size 40-x. The Kruskal-Wallis test revealed statistical differences in the means for large instances of PECEW (20-x and 40-x).




(a) PECEW 10-x (b) PECEW 20-x (c) PECEW 40-x

Fig. 1: Statistical summary of the fitness means obtained by applying different metaheuristic algorithms to several sizes of the problem PECEW.

In general, hyperheuristics (H) significantly improve the results of applying pure metaheuristics (P-M) and also improve the fitness resulting from the application of the best known hybrid metaheuristic (B-M). The metaheuristics obtained from hyperheuristics (M-H) offer good quality results in many cases, comparable with those reached with the best hybrid metaheuristic.

Future research could apply the hyperheuristic methodology to other optimization problems, even more widely studied ones, to compare our approach with other hyperheuristic implementations.

## Acknowledgements

This work was supported by the Spanish MINECO, as well as European Commission FEDER funds, under grant TIN2012-38341-C04-03.

## References

1. F. Almeida, D. Giménez, J.-J. López-Espín, and M. Pérez-Pérez. Parameterised schemes of metaheuristics: basic ideas and applications with Genetic algorithms, Scatter Search and GRASP. *IEEE Trnsactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 43(3):570–586, 2013.
2. E. Burke, E. Hart, G. Kendall, J. Newall, P. Ross and S. Schulenburg. Hyper-heuristics: An Emerging Direction in Modern Search Technology. In *Fred Glover and Gary Kochenberger, editors, Handbook of Meta-heuristics*, chapter 16, pages 457–474, 2003.
3. L.-G. Cutillas-Lozano, J.-M. Cutillas-Lozano, and D. Giménez. Modeling shared-memory metaheuristic schemes for electricity consumption. In *Distributed Computing and Artificial Intelligence - 9th International Conference*, pages 33–40, 2012.