Parallel systems are formed by multicore ⇒ develop efficient multicore versions of our algorithms, with OpenMP. A unified parameterised scheme for metaheuristics, facilitates the easy development of new metaheuristics or hybrid metaheuristics, for experiment and adaptation to a particular problem:

Initialize(S) while (not EndCondition(S))  
SS = Select(S)  
if(|SS| > 1) SS = Combine(SS)  
else SS = SS  
SS = Improve(SS)  
< Include(SS)

To obtain a well-tailored metaheuristic for a problem, experiment with a large number of metaheuristics and their parameters: the time dedicated to the experiments is very large ⇒ parallel programming. Common development of parallel versions: from the unified metaheuristic scheme to obtain a unified parallel scheme for metaheuristics.

The parallel scheme is parameterised: the values of some algorithmic parameters can be selected to optimise the execution of the parallel metaheuristic. The optimum values of the algorithmic parameters depend on those of the metaheuristic parameters and of the characteristics of the computational system.

Each basic function in the unified metaheuristic scheme can be parameterised: different values of the parameters give different metaheuristics, hybridization/combination of metaheuristics or different versions of a particular metaheuristic.

The number of threads in the first level (level1-thr) is obtained as a function of the parameters of the metaheuristic (also of its functions, and consequently of the cost of them in the computational system).

The number of threads to work in the second level (level2-thr) is obtained as a function of the metaheuristic parameters and the number of threads working in the first level.

Problem used in the experiments: given a set of data (obtained by experimentation, survey...) to obtain the best Simultaneous Equation Model (SEM) which best represents the variables dependences. SEM are developed by experts with a wealth of experience in the particular problem represented by the model.

An automatic tool to provide the experts with a wealth of experience in the particular problem represented by the model.

The number of threads to work in the second level (level2-thr) is obtained as a function of the metaheuristic parameters and the number of threads working in the first level.

The figure shows the speed-up achieved in the two systems with the seven metaheuristics:

- 8 or 128: using the maximum number of cores in the systems (8 in Arabi and 128 in Ben) without nested parallelism.
- lowest: with the number of cores in each parallelism level which gives the lowest execution time.
- parts: with the best combination of threads in the initialisation part and in the iteration part.

Ben is a HP Integrity Superdome SX2000 with 128 cores of the processor Intel Itanium-2 dual-core Montvale.

Arabi is a cluster of 102 nodes, each one with 8 cores of the processor Intel Xeon Quad-Core L5450.

Experiments in individual nodes: 8 and 128 cores.