20th Euromicro International Conference on Parallel, Distributed, and Network-based Processing PDP 2012
Garching, Germany
February 15-17, 2012

Analysing the Adaptation Level of Parallel Hyperheuristics applied to Multiobjectivised Benchmark Problems

Carlos Segura, Eduardo Segredo, Coromoto León
Dpto. Estadística, I.O. y Computación
Universidad de La Laguna

Outline

Metaheuristics: Evolutionary Algorithms

- A wide variety of approximation algorithms has been designed to solve optimisation problems.
- Metaheuristics are a family of approximation algorithms.
- Among them, *Evolutionary Algorithms* (EAs) are one of the most widely used strategies.
- To define a configuration of an EA several components, as the survivor selection mechanism and the genetic operators must be specified.
- Therefore, the process of making the parameter setting of an EA takes too much user and computational effort.

Multiobjectivisation

- *Multiobjectivisation* was introduced to refer to the reformulation of mono-objective problems as multi-objective ones.
- Multiobjectivisation has been used as a mechanism to avoid premature convergence to local optima.
- Multiobjectivisation might change the fitness landscape of the considered problem.
- Multiobjectivisation might add more parameters to the whole optimisation scheme.
- This hinders the proper usage of EAs.

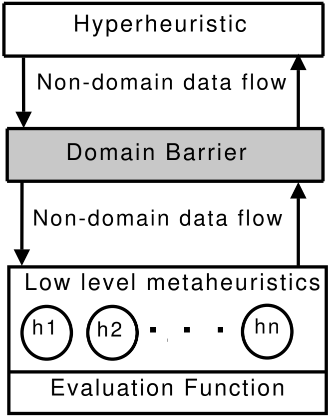
Hyperheuristics

- Hyperheuristics are a promising approach to facilitate the application of EAs.
- A hyperheuristic can be viewed as a heuristic that iteratively chooses between a set of given low-level (meta)-heuristics in order to solve an optimisation problem.
- Hyperheuristics can be tuned by the *adaptation level*.

Main Features

- They have no knowledge about the problem domain.
- They can provide high quality solutions in a single run.
- They can discover which are the most suitable low-level (meta)-heuristics for the considered problem.

Hyperheuristic Framework



Parallel EAs and Parallel Hyperheuristics

- In order to reduce the computational time invested by EAs, several studies have considered their parallelisation.
- *Parallel Evolutionary Algorithms* (pEAs) can be classified in three paradigms: *master-worker*, *island-based*, and *diffusion*.
- Parallel hyperheuristics have been proposed with the aim of reducing the gap between tailor-made schemes and hyperheuristic-based strategies.
- In previous works, a dynamic-mapped island-based model (DYN) has been applied.
- This model is a hybridisation which combines:
  - Parallel island-based models
  - Hyperheuristics

Main Contribution

- The main contribution is to analyse the adaptation level of the hyperheuristic incorporated into the dynamic-mapped island-based model.
- Such an analysis has been performed with a set of well-known benchmark problems.
- Such problems have been multiobjectivised.
- Thus, the second objective is to explore the advantages and drawbacks of incorporating multiobjectivisation in the parallel optimisation scheme.
- Results have been compared with another parallel approach that does not apply multiobjectivisation.

Applied Multiobjectivisation Approaches

Distance to the Closest Neighbour (DCN)

- The artificial objective function is calculated as the Euclidean distance to the closest member of the population, considering the genotype space.
- The minimisation of the original objective function and the maximisation of the artificial one are assumed.

Distance to the Closest Neighbour with Threshold (DCN-THR)

- It starts from the DCN function, but it incorporates the usage of a threshold percentage ($th \in [0, 1]$) which must be specified by the user.
- The threshold percentage is used to avoid the survival of individuals with a very low quality.

MultiObjective Evolutionary Algorithms

- Since multiobjectivisation methods have been used, a MultiObjective Evolutionary Algorithm (MOEA) must be considered.
- The well-known *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) has been applied.
- In order to apply this algorithm, a set of components and parameters must be specified.
- To obtain every low-level approach, different NSGA-II configurations have been combined with different multiobjectivisations.

Configurations (I)

- Specifically, 24 configurations have been considered by combining 3 crossover operators, 2 mutation operators, and 4 different multiobjectivisations.

Crossover Operators (applied with a probability p_c)

- *Simulated Binary Crossover* (SBX)
- *Uniform Crossover* (UX)
- *One-Point Crossover* (OPX)

Mutation Operators (applied with a probability p_m)

- *Polynomial Mutation* (PM)
- *Uniform Mutation* (UM)

Configurations (II)

Multiobjectivisations

- DCN
- DCN-THR with $th = 0.1$
- DCN-THR with $th = 0.5$
- DCN-THR with $th = 0.99$

Other components

- Candidate solutions have been represented by a vector of D real numbers, being D the number of variables of the considered benchmark problem.
- Finally, the parent selection mechanism has been the well-known *Binary Tournament*.

Architecture (I)

- The architecture is similar to the parallel island-based model.
- It is constituted by a set of *worker islands* which evolve in isolation applying a certain low-level configuration to a given population.
- A tunable migration stage allows the exchange of individuals between neighbour islands.
- In the considered model, a dynamic mapping among the islands and configurations is applied, i.e., configurations executed on each island along the run can vary.
- Such a mapping is performed using a hyperheuristic.
- A new special island, called *master island* is introduced into the architecture so as to manage the dynamic mapping.

Architecture (II)

- In the standard island-based model, a global stopping criterion is defined.
- When this global stopping criterion is reached, every worker island sends its local solution to the master and the run ends.
- In the DYN model, besides the global stopping criterion, local stopping criteria are fixed for the execution of the configurations on the worker islands.
- When a local stopping criterion is reached, the island execution is stopped, and local results are sent to the master island.
- At this point, the master island applies the hyperheuristic in order to decide which low-level configuration will be applied in the idle island.

Hyperheuristic

- The incorporated hyperheuristic (HH_imp) is based on using a scoring strategy and a selection strategy.
- The scoring strategy assigns a score $s(conf)$ to each configuration *conf* which has been executed j times.
- This score estimates the improvement *imp* that each configuration can achieve when it starts from the currently obtained solutions.
- In order to perform such an estimate, the previous k improvements achieved by each configuration are used.
- k allows to define the adaptation level of the hyperheuristic.
- Being n_h the number of involved low-level configurations, a random selection following a uniform distribution is performed in $\beta * n_h$ percentage of the cases.

Hyperheuristic: Formal Definition

$$prob(conf) = \beta + (1 - \beta * n_h) * \left[\frac{s(conf)}{\sum_{i=1}^{n_h} s(i)} \right]$$

$$s(conf) = \frac{\sum_{i=1}^{min(k,j)} (k + 1 - i) * imp[conf][j - i]}{\sum_{i=1}^{min(k,j)} i}$$

Computational Resources

- Tests have been run on a machine with the following characteristics:
  - Debian GNU/Linux
  - 4 AMD ® Opteron ™ (6164 HE) at 1.7 GHz
  - 64 GB RAM
- The compiler has been GCC.
- The MPI library has been MPICH 1.2.7.
- Metaheuristic Extensible Tool for Cooperative Optimisation (METCO) has been used to implement the optimisation schemes.

Parameterisation

- Different experiments have been applied to the well-known F1-F11 mono-objective benchmark problems.
- A total number of $D = 500$ variables has been fixed.
- The population size n has been fixed to 5 individuals.
- For all experiments, the following parameterisation has been used: $p_c = 1$, and $p_m = \frac{1}{D}$.
- Every execution has been repeated 30 times.
- Statistical tests have been performed using a confidence level of 95%.
- In every experiment, the number of worker islands (n_p) has been fixed to 4.

First Analysis

- **Objective**: Analysing the adaptation level of the hyperheuristic of the dynamic-mapped island-based model (DYN).
- The value β has been fixed in a way that the 10% of the decisions performed by the hyperheuristic of the DYN model follows a uniform distribution, i.e. $\beta \cdot n_h = 0.1$.
- The DYN model has been executed with several values of k: 5, 10, 50, 100, 500, and ∞.
- In addition, a dynamic-mapped island-based model that assigns the resources in a uniform way (UNI) among the low-level configurations has also been executed.
- The global and local stopping criteria have been fixed to $1 \cdot 10^7$ and $1 \cdot 10^3$ evaluations, respectively.

Percentage of Saved Evaluations with the DYN Model

	F1	F2	F3	F4	F5	F6
k = 5	57.14 %	-9.27 %	56.42 %	56.16 %	43.85 %	63.80 %
k = 10	51.70 %	-0.66 %	54.28 %	52.05 %	31.57 %	58.28 %
k = 50	25.85 %	2.64 %	42.14 %	24.65 %	15.78 %	35.58 %
k = 100	8.16 %	7.28 %	26.42 %	6.84 %	-10.52 %	20.24 %
k = 500	-28.57 %	4.63 %	-12.14 %	-20.54 %	-14.03 %	3.68 %
$k = \infty$	61.90 %	-31.78 %	57.14 %	57.53 %	40.35 %	66.25 %

	F7	F8	F9	F10	F11
k = 5	62.4 %	30.15 %	44.67 %	60.25 %	47.97 %
k = 10	56 %	28.64 %	46.70 %	54.48 %	49.49 %
k = 50	24.8 %	21.60 %	32.48 %	28.20 %	35.35 %
k = 100	-0.8 %	17.08 %	25.38 %	9.61 %	24.74 %
k = 500	52 %	2.01 %	9.13 %	-27.56 %	8.58 %
$k = \infty$	66.4 %	29.64 %	30.96 %	64.10 %	31.31 %

Speedup of the DYN Model

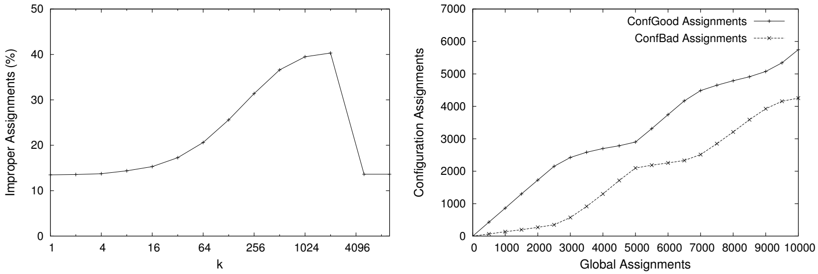
	F1	F2	F3	F4	F5	F6
k = 5	4.59	4.55	4.14	7.15	4	5.36
k = 10	4	4.65	3.5	6.09	3.36	4.62
k = 50	2.7	4.82	2.8	4.27	2.72	2.95
k = 100	2.37	4.95	2.24	3.72	2.06	2.45
k = 500	1.87	4.82	2.07	3.35	2	2.31
k = ∞	5.4	3.32	4.66	6.88	4	6.05

	F7	F8	F9	F10	F11
k = 5	3.02	2.16	3.88	4.43	3.94
k = 10	2.54	2.19	3.67	3.84	3.94
k = 50	1.51	1.93	2.89	2.42	3.11
k = 100	1.19	1.78	2.56	2.17	2.66
k = 500	1.53	1.50	2.45	1.71	2.52
k = ∞	3.61	2.19	3.62	5.23	3.78

Second Analysis

- **Objective**: Studying the reasons of the bad behaviour of the hyperheuristic with intermediate values of k.
- A trivial optimisation problem and two artificial low-level configurations have been considered.
- Problem: Minimise $f(x_1) = x_1$.
- Low-level configurations:
  - *ConfGood*: $newx_1 = x_1 - \frac{x_1}{100}$
  - *ConfBad*: $newx_1 = x_1 - \frac{x_1}{1000}$
- *ConfGood* behaves better than *ConfBad* for any input.
- The best assignment of resources would consist on granting all the resources to *ConfGood*.

Assignments for the Trivial Problem



Third Analysis

- **Objective**: Analysing the contribution of the incorporation of multiobjectivisation approaches to the DYN model.
- Such a model has been compared with a similar parallel hyperheuristic that does not incorporate multiobjectivisation.
- The low-level metaheuristics are similar, but instead of using the NSGA-II, a mono-objective EA has been applied.
- Specifically, 18 different configurations, which combine the same genetic operators here presented with three different survivor selection operators, have been defined.
- The survivor selection operators have been the following ones: *steady-state*, *generational with elitism*, and a *replace-worst* operator.

Median of the Error in $2.5 \cdot 10^6$ Evaluations

	F1	F2	F3	F4	F5	F6
Mono, $k = 5$	0.058	29.69	1550.81	0.042	0.010	0.013
Multi, $k = 5$	0.001	10.81	1077.2	$< 1 \cdot 10^{-6}$	$< 1 \cdot 10^{-6}$	$< 1 \cdot 10^{-6}$
Mono, $k = \infty$	0.026	28.36	1459.44	0.010	0.004	0.007
Multi, $k = \infty$	0.0005	16.64	1081.7	$< 1 \cdot 10^{-6}$	0.001	$< 1 \cdot 10^{-6}$

	F7	F8	F9	F10	F11
Mono, $k = 5$	0.315	$1.525 \cdot 10^{11}$	112.92	0.07	112.01
Multi, $k = 5$	0.002	$1.033 \cdot 10^{11}$	47.70	$5.9 \cdot 10^{-4}$	46.12
Mono, $k = \infty$	0.181	$1.512 \cdot 10^{11}$	102.11	0.03	103.78
Multi, $k = \infty$	0.002	$1.029 \cdot 10^{11}$	54.46	$8.1 \cdot 10^{-4}$	54.77

Conclusions

- In most cases, the DYN model, using low or very high values of k, has been able to save more than a 50% of evaluations when compared to the UNI model.
- The speedup analysis has also revealed the good behaviour of the DYN model with such values of k.
- The reasons of the bad behaviour of the DYN model using intermediate values of k have also been explored.
- The superiority of the DYN model with multiobjectivisation has been demonstrated when it has been compared to the parallel model without multiobjectivisation.
- The median of the error obtained by such a model has been lower for every benchmark problem, and the statistical tests have confirmed its superiority.

Future Work

- Future work targets the improvement of the DYN model when it is applied with intermediate values of k.
- This would improve the robustness of the model in terms of the adaptation level.
- Currently, the hyperheuristic works with a set of prefixed configurations.
- A great line of research would be to develop a hyperheuristic capable of performing the parameter setting without the requirement of specifying such a set of configurations.

Questions?

Thank you for your attention!

Acknowledgements

This work has been partially supported by the EC (FEDER) and the Spanish Ministry of Science and Innovation as part of the 'Plan Nacional de I+D+i', with contract number TIN2011-25448. The work of Carlos Segura has been funded by grant FPU-AP2008-03213. The work of Eduardo Segredo has been funded by grant FPU-AP2009-0457.