Heterogeneous Island Model with Re-planning of Methods

Štěpán Balcar
Charles University, Faculty of Mathematics and Physics
Prague, Czech Republic
Stepan.Balcar@mff.cuni.cz

Martin Pilát
Charles University, Faculty of Mathematics and Physics
Prague, Czech Republic
Martin.Pilat@mff.cuni.cz

ABSTRACT

We propose a heterogeneous island model where each of the islands can run a different optimization algorithm. The distributed computation is managed by a central planner, that re-plans the methods during the run of the algorithm – less successful methods are removed while new instances of more successful methods are added. We show that this re-planning improves the performance of the optimization algorithm.

CCS CONCEPTS

• **Computing methodologies → Parallel algorithms**; Artificial intelligence; • **Computer systems organization → Distributed architectures**;

KEYWORDS

Parallel evolutionary algorithms; island model; algorithm selection; iterative re-planning; adaptive system.

ACM Reference Format:
Štěpán Balcar and Martin Pilát. 2018. Heterogeneous Island Model with Re-planning of Methods. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205786

1 INTRODUCTION

Many optimization methods based on various paradigms have been proposed. some of them are better for one problem, while others are better for another. In fact, the most suitable method can even be different in different phases of the optimization. So called portfolio algorithms [1] run multiple independent instances of different algorithms in parallel (or sequentially on a single CPU) with the goal to find better solution. Quite recently, Lindenauer *et al.* [4] studied the problem of portfolio selection as an extension of the problem of algorithm selection.

We present an online algorithm for parallel portfolio selection. It is based on a heterogeneous island model derived from the homogeneous island model commonly used for the parallelization of evolutionary algorithms. The island model [2] is based on the idea of several isolated populations evolving in parallel. The islands exchange the individuals from their populations, thus cooperating with each other and accelerating the convergence. Our model runs a number of different stochastic optimization methods that periodically exchange some of the solutions they found. Additionally, there is also a planner that adaptively replaces the under-performing methods by better-performing ones and thus optimizes the set of algorithms used in the model online.

2 HETEROGENEOUS ISLAND MODEL

We assume a parallel computational environment with multiple CPU cores. Each CPU core corresponds to a single island and each island then executes a single optimization method. The methods running on different islands only have to share the encoding of the individuals, but otherwise can be different.

The performance of the various methods is monitored during the optimization and various metrics are evaluated, including how often each methods finds new best solution, or how often the solution found is among the top ones. This information is then used by a planner that controls the whole system and attempts to improve its performance by changing the set of methods running at any given time. The planner first initializes the islands with different methods in a round-robin manner. In case there are more different types of methods than the number of islands, methods that were not used in initialization are given priority later.

Afterwards, the planner periodically checks the information about the performance of the methods and removes the worst performing one, replacing it with the best performing one. Here, "worst" and "best" are defined by the particular planner and will be described bellow. In case some of the available methods has not been executed yet, it is automatically selected as the best one.

The various optimization methods run completely independently and in a fully asynchronous way. They periodically send their best solutions to the other running instances and also receive new solutions from them. Therefore, the common optimization methods must be slightly modified before they can be used in the heterogeneous island model. For the tested methods, new solutions are added as if they were generated by the method, e.g. in hill-climbing the solution is accepted if it is better than the current best solution for the method, and in evolutionary algorithms, the received solutions are added to the population.

We present two different planners in this paper, they both follow the procedure outlined in Algorithm 1. The *Quantity of Improvement (P-QI)* planner uses the number of times each method produced the currently best solution. In each planning iteration, the method that produced the least number of best solutions since the last re-planning is replaced by the method that produced the most such solutions. Newly started methods are protected for the first $N_{protect} = 3$ planning iterations.

The *Best Material (P-BM)* planner is based on the number of times each method provided a solution among the top N solutions overall. In each planning iteration, the planner removes the method

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205786

Algorithm 1 General overview

```
Initialize the methods uniformly on the islands
t ← 0
while termination condition not met do
    I ← obtain information about running methods
    if there is a method M that has not run then
        k ← the least useful method
        s ← M
    else
        k ← select a method to kill using a planner-spec. rule
        s ← select a method to start using a planner-spec. rule
    end if
    Kill method k and start method s
    t ← t + 1
    Sleep until next planning iteration
end while
```

with the least number of top solutions since the last re-planning and replaces it with the method with the most top solutions.

3 EXPERIMENTS

In order to test the proposed heterogeneous island model, we ran experiments where we compare the different planners to homogeneous island model running each of the optimization methods and to heterogeneous island model running all the methods, each in two instances, without re-planning. The experiments were performed on two instances of the traveling salesman problem (TSP) (with 1083 and 662 cities) from VLSI[1] and one randomly generated instance of the bin-packing problem (BPP) (uniformly random numbers from the interval [0,1] are packed into bins of size 1).

In all the experiments, the number of planning iterations is set to 50, each iteration takes 60 seconds and the experiments are run on 16 CPUs (islands). We have implemented seven different optimization methods – random search (RS), tabu search (TS), hill climbing (HC), simulated annealing (SA), evolutionary algorithm (EA), differential evolution (DE), and a brute force (BF) algorithm that performs a systematic search. Using DE for these problems is not very typical, but we also did experiments in continuous optimization (not presented here due to space limitations) and wanted to have the set of algorithms consistent. The HC looks for 10 random neighbors in each iteration, EA uses population of 10 individuals, binary tournament, and mutation and crossover rate of 0.9 and 0.1 respectively. The TS has a tabu set size of 50. The SA starts with temperature 10,000 and uses cooling rate of 0.002. The DE has population size of 50 individuals and the parameter $F = 1$ (crossover is not used). Both the problems use permutation encoding, therefore the operators are the same for RS, BF, and DE – the RS generates new permutations randomly, the BF generates the next permutation in lexicographic order and the DE computes the pair-wise difference of the permutations, adds it to a third one and generates the new permutation by sorting the indices of the resulting vector by the values in the vector. The rest of the methods work differently for each problem. For TSP, the HC, TS and SA use the 2-opt operator [3] to generate new solutions.

[1] http://www.math.uwaterloo.ca/tsp/vlsi/

Table 1: Results of the planners. The numbers represent the average value of the objective over nine independent runs. The numbers in subscript and superscript show the minimum and maximum value of the objective achieved in the experiments.

Method	TSP1083	TSP662	BPP1000
Best homogen.	5154^{5221}_{5084}	2997^{3073}_{2946}	521.7^{524}_{520}
Hetero - Static	5253^{5313}_{5105}	3055^{3102}_{2994}	529.2^{535}_{527}
Hetero - P-BM	4985^{5035}_{4912}	3014^{3062}_{2947}	523.2^{527}_{520}
Hetero - P-QI	5022^{5098}_{4951}	3005^{3080}_{2960}	525.2^{532}_{522}

The EA uses the same operator as a mutation and a single-point crossover with repair of the permutation. For the BPP, the HC, TS, and SA use a displacement operator that moves a randomly selected number (less than 0.5 per cent of the length of the individual) of consecutive values to the end of the permutation. The EA uses the order crossover and a shift mutation that moves a random object to the end of the permutation.

The results of the experiments are provided in Table 1. We can see that the heterogeneous island model provides better or comparable results compared to the best method executed in the homogeneous island model. However, finding the best method in the homogeneous setting requires running all the methods, which is extremely time-consuming. The heterogeneous implementation has similar performance in a single run. For the TSP1083 problem, the P-BM provided the best result, followed by P-QI, the best homogeneous method (HC in this case) and the static heterogeneous model. In the TSP662 experiment, the best homogeneous method (TS) provided slightly better result (2997) than the heterogeneous methods (3005 for the P-QI). Similarly, for the BPP test, the best homogeneous method (EA) is only slightly better than the heterogeneous ones (521.7 for EA vs 523.2 for P-BM and 525.2 for P-QI).

We can therefore conclude, that if multiple CPU cores are used for optimization and the best optimization method is not known beforehand (which is the typical case), it is better to use the heterogeneous model that selects the method automatically than using multiple runs with the homogeneous model. The heterogeneous models can thus also be considered an algorithm selection method.

ACKNOWLEDGMENT

Martin Pilát was supported by Czech Science Foundation project number 17-17125Y and Štěpán Balcar was supported by SVV project number SVV-2018-260451.

REFERENCES

[1] Carla P. Gomes and Bart Selman. 2001. Algorithm portfolios. *Artificial Intelligence* 126, 1 (2001), 43 – 62. https://doi.org/10.1016/S0004-3702(00)00081-3

[2] J. Kacprzyk and W. Pedrycz. 2015. *Springer Handbook of Computational Intelligence*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43505-2

[3] Shen Lin. 1965. *Computer solutions of the traveling salesman problem*. The Bell System Technical Journal. 2245 – 2269 pages. https://doi.org/10.1002/j.1538-7305.1965.tb04146.x

[4] M. Lindauer, H. Hoos, and F. Hutter. 2015. From Sequential Algorithm Selection to Parallel Portfolio Selection. In *Learning and Intelligent Optimization*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-19084-6_1