Towards a Novel NSGA-II-based Approach for Multi-objective Scientific Workflow Scheduling on Hybrid Clouds

Haithem Hafsi
National School of Computer Science (ENSI),
Manouba University
Manouba, Tunisia
haithem.hafsi@gmail.com

Hamza Gharsellaoui
Khurmah University College (KUC),
Taif University
Taif, KSA
gharsellaoui.hamza@gmail.com

Sadok Bouamama
Higher College of Technology (DMC)
Dubai, United Arab Emirates
Sbouamama@hct.ac.ae

ABSTRACT

In the era of the big data and e-science revolution, the execution of such applications known as High Performance Computing (HPC) is becoming a challenging issue. In order to face these challenges, a new promising Large Scale Distributed Systems (LSDS) has emerged suchlike Grid and Cloud Computing. As a matter of fact, these HPC applications are commonly arranged as a form of interdependent tasks named workflows. Nevertheless, the new challenging topic is that the scheduling of these scientific workflows in the LSDS is a well-known NP-hard problem. The goal of this work is to design an Non-dominated Sorting Genetic Algorithm Version II (NSGA-II)-based approach for optimizing a multi objective scheduling of scientific workflows in hybrid distributed systems. This paper work deals with the proposition of two execution models: i) A *Cumulative* one aiming to improve the Pareto front quality in term of Makespan-Cost trade-off; ii) An *Incremental* execution fashion, what kind of Cost-driven approach leading to a solution diversity of the Pareto front in the objective space. Experiments conducted with multiple common scientific workflows point out significant improvement against the classic NSGA-II algorithm.

KEYWORDS

Multi-objective optimization, hybrid clouds, workflow scheduling, NSGA-II.

ACM Reference Format:

Haithem Hafsi, Hamza Gharsellaoui, and Sadok Bouamama. 2019. Towards a Novel NSGA-II-based Approach for Multi-objective Scientific Workflow Scheduling on Hybrid Clouds. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321975

1 INTRODUCTION

Evolutionary Multi-objective Optimization (EMO) algorithms are the most commonly adopted methods to search for the optimal trade-off between objectives in the Multi-objective Optimization Problem (MOP). For the workflow scheduling problem, a significant diversity of EMO algorithms were designed [2] [4] [3]. As long as it is widely used for optimizing MOPs, we adopted the Non-dominated Sorting Genetic Algorithm Version II (NSGA-II) [1] algorithm to optimize scientific workflow scheduling in hybrid computing infrastructures considering Makespan and Economic Cost as objectives. Actually, we have proposed a typical encoding better-representing the scheduling solution on a hybrid computing infrastructure obtained by extending private resources with Cloud virtual machines. In addition we have designed two execution models of the NSGA-II algorithm. These models consist of dividing the hole NSGA-II execution into *Bloc*s in which the initialization routine is re-processed in a specific manner.

2 PROPOSED APPROACH

Basing on the NSGA-II algorithm that we noted as *Standard*, we designed two execution models named *Cumulative* and *Incremental* described as follows:

Our proposed *Cumulative* execution model consists of dividing the total iteration number Ng into $Nrep$ Standard NSGA-II *Bloc*s. The Pareto front resulted by executing the i^{th} *Bloc* will be completed by initializing the missing solutions to get the initial population of the *Bloc* $(i+1)$. This operation is repeated like so until executing an overall Ng iterations. The idea is to make a refresh of generations by repeating the initialization process and by conserving the obtained non-dominated front in each *Bloc*. By designing this model, we don't aim to preserve the found Pareto front because the elitism aspect of NSGA-II does, but the novelty is to refresh the population by a new initialized solutions or individuals to improve the descending of the new population.

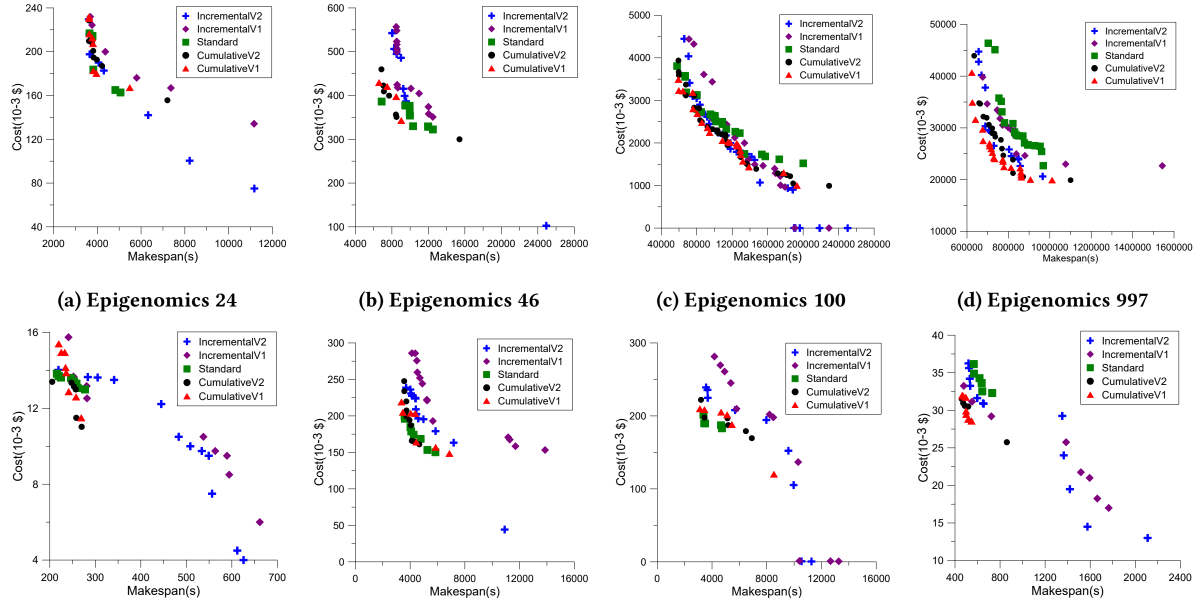
On the other hand, the *Incremental* model is based on the same idea, except that we consider a different set of *VM*s in the initialization process in each *Bloc*. In fact, while *Cumulative* model searches scheduling solutions basing on all paid machine images from the beginning, the *Incremental* model append them incrementally by adding the lower priced *VM*s then the higher ones in each *Bloc*. In such cost-driven approach, we try to determine the best solutions for different hybrid resource sets by increasing the execution budget.

3 EXPERIMENTATION AND DISCUSSION

To evaluate our proposed approaches, different simulation scenarios was carried out on 5 types of synthetic workflows with various number of workflow nodes. So that we make this comparison, we

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321975



(a) Epigenomics 24 (b) Epigenomics 46 (c) Epigenomics 100 (d) Epigenomics 997

(e) Montage 100 (f) Inspiral 100 (g) Sipht 100 (h) CyberShake 100

Figure 1: Results with Different Workflow types

fixed the total number of iterations to 2500 for all the scenarios and a population size to 100. In fact, we note *CumulativeV1* a simulation that performs 5 *Blocs* of 500 iterations while the *CumulativeV2* executes 25 *Blocs* of 100 iterations for each. On the other hand, the *Incremental* model will be realized also in two ways : the first one named *IncremetalV1* in which we operate 5 *Blocs* of 500 iterations. The second way, noted *IncrementalV2*, contains also 5 *Blocs*. But here, each *Bloc* of 500 iterations will be carried out in *Cumulative* way. Finally, the *Standard* NSGA-II will be run with 2500 successive iterations as a single *Bloc*.

As illustrated in Figure 1 and in most cases, the *Cumulative* model outperforms the rest of execution models in term of Pareto dominance. In addition, good results of the *Cumulative* model are especially presented by the *CumulativeV1* scenario which slightly better than the *CumulativeV2* one.

Taking the Epigenomics workflow as an example of application, we will discuss the manner how our proposed approaches change by increasing the number of workflow nodes and the task load. As we can see in Figures 1a, 1b, 1c and 1d, the *Cumulative* model with its two versions gives a solution fronts more and more better than *Standard* model as long as we increase the node number from 24 to 997 nodes and especially for heavy tasks load in Figures 1c and 1d in which we achieve a gain up to 24.8% in time against *Standard* model.

Although *Incremental* model performance was not perfect in some simulations in terms of Pareto dominance, we still believe that it presents some advantages and these results need to be interpreted with caution. In fact, this model was designed to find solutions with lower budgets, for that we get a part of solutions show lower costs and relatively higher makespan as we can see especially for Montage and CyberShake workflows in Figures 1e and 1h. The lower cost is evidently explained by the lower number of paid resources in different *Bloc*s while the noteworthy higher makespan is simply explained by our choice of the free resources' configuration which is lower than paid ones.

To sum up, the called *Cumulative* model aims to enhance the solution quality by initializing new solutions in addition to the generated Pareto front after each *Bloc* of iterations. This model proves its efficiency regards to the standard NSGA-II as experiments show. The second approach named *Incremental* was realized in order to lease VM progressively (in each *Bloc*) trying to minimize execution time and controlling the increase of cost. This cost-driven execution fashion presents good results according to the solutions diversity but needs to be improved to generate more optimal solutions in terms of Pareto dominance.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (April 2002), 182–197. https://doi.org/10.1109/4235.996017
[2] Nadia Nedjah and Luiza de Macedo Mourelle. 2015. Evolutionary Multi-objective Optimisation: A Survey. *Int. J. Bio-Inspired Comput.* 7, 1 (March 2015), 1–25. https://doi.org/10.1504/IJBIC.2015.067991
[3] Poonam, Maitreyee Dutta, and Naveen Aggarwal. 2016. Meta-Heuristics Based Approach for Workflow Scheduling in Cloud Computing: A Survey. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Subhransu Sekhar Dash, M. Arun Bhaskar, Bijaya Ketan Panigrahi, and Swagatam Das (Eds.). Springer India, New Delhi, 1331–1345.
[4] Zhi-Hui Zhan, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. 2015. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *ACM Comput. Surv.* 47, 4, Article 63 (July 2015), 33 pages. https://doi.org/10.1145/2788397