Impact of Voltage Levels Number for Energy-aware Bi-objective DAG Scheduling for Multi-processors Systems

Mateusz Guzek[1], Cesar O. Diaz[2], Johnatan E. Pecero[2], Pascal Bouvry[2], Albert Y. Zomaya[3]

[1] Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, mateusz.guzek@uni.lu
[2] Computer Science and Communications Research Unit, University of Luxembourg {firstname.lastname}@uni.lu
[3] Centre for Distributed and High Performance Computing, The University of Sydney albert.zomaya@sydney.edu.au

Abstract. The paper investigates the influence of Dynamic Voltage Frequency Scaling for bi-objective (makespan and energy consumption) Directed Acyclic task Graph scheduling on heterogeneous multi-processor platform. The proposed resolution method of solving conflicting criteria relies on Multi-Objective Evolutionary Algorithms. Two voltage frequency approaches are compared: one using only 2 levels (minimal and maximal) and the other one using a larger number. The approaches are benchmarked on applications with Laplace transformation and Gaussian elimination structures using the NSGAII algorithm. The results show that while the Two-level approach generates more discriminated solutions on the Pareto front, the solutions are of a lower quality than in the Multi-level approach.

1 Introduction

Energy consumption of computing systems became the main operational expense and it is still growing. A recent study reports 31 GW of global data center power consumption and estimates 19% increase by 2012 [1]. The above increase has to be tempered to meet constraints given by the environment, the society and the industry.

Various techniques to manage efficiently the energy consumption in processors have been developed by hardware designers, among which two can be directly used at the level of scheduling: Dynamic Power Management (DPM) technique and Dynamic Voltage Frequency Scaling (DVFS). DPM consolidates applications on a minimum set of computing resources to maximize the number of resources that can be powered down while maximizing utilization of the used ones [2]. DVFS enables processors to dynamically change the working supply voltage and clock speed without stopping or pausing the execution of any instruction. The technique is developed under premise that in CMOS technology

there is a range of voltage where the CMOS transistors work in the operational mode. DVFS reduces energy consumption of processors based on the fact that power consumption in CMOS circuits has direct relation with frequency and the square of the supplied voltage. By lowering clock speed and supply voltage during frequency-insensitive application phases, large reductions in power can be achieved with modest performance loss [3]. In this paper a scheduler based on a Multi Objective Evolutionary Algorithm (MOEA) uses DVFS. The representation of the solution is extended by a part that indicates a DVFS level selected for task execution.

The impact of the aforementioned extension on the search process is investigated in this paper. The research question is therefore: does increasing the granularity of available voltage levels increase the quality of results despite the exponential growth of the search space? We provide experimental evidence that increase of available voltages number in fact increases the quality of solutions, but reduces their number. We verify the hypothesis by using a state-of-the-art multi-objective evolutionary algorithm NSGAII. The approaches are benchmarked on applications with Laplace transformation and Gaussian elimination structures. As a result, quality indicators: Additive Epsilon, IGD, and Spread, are computed and analyzed by Wilcoxon and Friedman statistical tests. Additionally, the average number of returned solutions is used for comparison.

The remainder of the paper is structured as follows: Section 2 presents the state-of-the-art scheduling algorithms that use DVFS technique. The studied models of system, application, energy, and scheduling are defined in Section 3. In Section 4 two MOEAs representations are proposed. Section 5 presents the results of experiments conducted to validate the hypothesis. In Section 6 we conclude the paper, discuss the consequences of the results analysis, and propose future directions of studies.

2 Related Work

An important number of scheduling algorithms have been proposed for energy consciousness. The most common technique is to combine DVFS with slack sharing or slack reclamation [4]. Different DVFS-based scheduling algorithms assume that the task assignment to processors is already done and then, the goal is to develop algorithms for the most appropriate voltage selection. The main idea is to distribute the slack among the tasks. The scheduling algorithm has to be voltage selection aware and try to maximize the possibility of saving computational energy in the voltage/frequency or speed selection step. In [5], Pruhs et al. consider the problem of scheduling DAGs with voltage scaling to satisfy the dual objectives of makespan and energy optimization. The authors show that the search space can be restricted to those with constant power schedules and show how to reduce the problem to develop a set of approximation algorithms for both objectives. However, the model does not consider communication delays.

In energy-conscious scheduling algorithms based on multi-objective approaches, the scheduling is computed with energy saving considerations to satisfy both

QoS and energy constraints simultaneously. The problem is modeled as a multi-constrained bi-objective optimization problem, and the goal is to find Pareto optimal schedules [4, 6, 7].

Some energy-aware scheduling algorithms are based on the *best-effort* idea. That is, first to optimize makespan, then voltage, frequency or speed scaling is performed. Rizvandi et al. [8] report a set of heuristics that operates in the two phases. Initially, a schedule that minimizes the makespan is found. The second phase tries to find the right setting of processors to minimize energy consumption without changing the length of the schedule. The proposed solution is called Multiple Voltage Frequency-Selection DVFS and compared against Maximum Minimum Frequency DVFS algorithm. The energy consumption is optimized by linear combinations of the two optimal, and minimum and maximum processor frequencies, respectively. Our work is different in two main aspects: we consider heterogeneous processors and the DVFS choice is done during schedule construction, not as a post-processing technique.

3 Problem Description

The target execution support we consider is a distributed computing system made up of a set R of m heterogeneous processors. Each processor $r_j \in R$ is DVFS-enabled; it can be operated on a set $DVFS_j$ of voltage and relative speed pairs. Each pair is a supply voltage v_k and the corresponding relative speed rs_k. Since clock frequency transition overhead takes negligible amount of time, the above overhead is not considered in this paper.

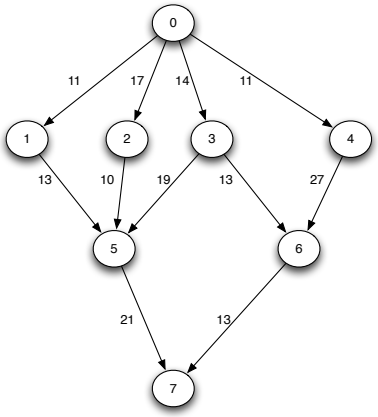
A parallel application is represented by a *Directed Acyclic Graph* (DAG). The DAG is defined as $G = (T, E)$, where T is a finite set of t nodes and E is a finite set of edges. The node $t_i \in T$ is associated with one task t_i of the modeled program. Each task $t_i \in T$ has an associated basic execution time which is an independent value for each machine. The basic execution time of task t_i on machine r_j at maximum speed and voltage is denoted as p_{ij}. Each edge $(t_i, t_j) \in E$ (with $t_i, t_j \in T$) is a precedence constraint between tasks and represents inter-task communications. The weight on any edge $(t_i, t_j) \in E$ stands for the communication time, denoted as $c_{i,j}$. However, a communication cost is only required when two tasks are assigned to different processors. Figure 1 shows a sample task graph.

The energy model is based on the CMOS energy equation. The dynamic power consumed by the CMOS circuit is expressed as [4]:

$$P = AC_{\mathrm{e}f}V^2 f, \tag{1}$$

where A is the number of switches per clock cycle, $C_{\mathrm{e}f}$ is the total capacitance load, V is supplied voltage, and f is the corresponding frequency.

DVFS pairs are sets of corresponding voltage and relative speed pairs and defined for each processor. As relative speed is proportional to operational frequency, we directly include it into our model in place of frequency to reduce error and get as close to the provided data as possible. $AC_{\mathrm{e}f}$ is set to 1 to abstract the



task	r_0	r_1	r_2	$\overline{p_i}$	b-level
0	11	13	9	11	101.3
1	10	15	11	12	66.7
2	9	12	14	12	63.3
3	12	16	10	12	73.0
4	15	11	19	15	79.3
5	13	9	5	9	41.7
6	11	15	13	12	37.3
7	11	15	10	12	12.0

Fig. 1: On the left a sample DAG with the task indexes i inside nodes and values of $c_{i,j}$ function next to the corresponding edges. On the right computation cost (p_i at level L_0) and task priorities (b-level).

results from the circuit architecture influence and normalize processors behavior. The final energy consumed by a system is defined by the following formula:

$$E_t = \sum_{j=0}^{m} \int_0^{C_{max}} P_j(t)dt, \tag{2}$$

where $P_j(t)$ is a function of power of processor j, defined as value of power function given by Equation 1 at moment t.

The scheduling problem is the process of allocating the set T of t tasks to the set R of m DVFS-enabled and unrelated processors, and assigning a starting time for each task to minimize the makespan and energy consumption. The makespan C_{max} of a schedule is the schedule length. The first task starts the execution at time 0, so that the makespan of a schedule is the maximum time at which one of the processors finishes its computation. As energy is a convex function of frequency, running a task using lower frequency results in smaller final energy consumption. The two objectives are in conflict, so different plausible solutions produce trade-offs between the objectives.

4 Compared Methods

MOEA [9] is one of the state-of-the-art multi-objective optimization approaches. Contrary to others, the algorithm does not aggregate objective functions into one function or give them priorities, but it explicitly tackles multiple objectives, using the concept of Pareto optimality. A tentative solution is called *Pareto optimal* when there is no other solution which can optimize one of the objectives without deteriorating the quality of other objective functions. The output of a MOEA is a set of Pareto optimal solutions called *Pareto set*. Mapping of the set to the corresponding objective space is called *Pareto front*.

To compare two different optimization approaches that use MOEA scheme we have chosen NSGAII [10] algorithm. It was motivated by its good performance and wide adoption in literature as a reference MOEA technique. NSGAII is based on the Genetic Algorithm (GA) schema. Each solution is represented as a chromosome (a sequence of genes). Each gene codes the value of one decision variable. The representation is used in two ways. Firstly it is used to create new solutions by application of operators: crossover, which combines parts of two existing solutions to create a new one, and mutation, which changes the existing solution. The representation is also used by evaluation function, which maps the encoded decision variables values into objective functions space.

The proposed representation for the bi-objective scheduling is based on the processors allocation representation [11] that codes the solution as a vector of task allocation data. Each position in the vector corresponds to one task and the associated value is a processor identifier. We extend the representation by adding the DVFS information, namely the selected DVFS pair, which is used for the task execution (see Figure 2). To finally establish the schedule, the list scheduling with the insertion technique and priorities based on b-level is applied. In case of existence of slacks in a schedule, the slack reclamation is used to combine the indicated in a gene DVFS pair with the minimum one. The idle time slots have allocated the minimum voltage.

Task Id	0	1	2	3	4	5	6	7
Processor	1	1	2	2	0	0	1	2
DVFS Pair	0	0	1	3	0	4	6	3

Fig. 2: A sample solution in extended processor assignment representation for a three-processor system with 5, 7 and 4 DVFS pairs available for consecutive processors. E.g. task 5 is assigned to processor 0 with DVFS pair 4.

The crossover operator used in the study is the grouping crossover, which has proven to outperform two-point crossover for similar problem and to intrinsically minimize communications [12]. The mutation is problem specific single-point mutation, which changes processor assignment and DVFS pair or only the DVFS pair with the same uniform probabilities.

Extending the representation results in vast expansion of the search space. In the basic processor allocation representation the search space has size m^t, where m is number of machines and t number of tasks. In the extended case, it is possible to assign a task to any processor using any DVFS pair of the selected processor. As a result the search space has size $(\sum_{i=0}^{m} levels_i)^t$, where $levels_i$ is the number of DVFS pairs available for machine i. We call this representation **Multi-voltage**. To mitigate the above effect we propose limited version that allows only the highest and the lowest voltages. In such case the search space has size of $(2m)^t$ and we call the representation **Two-voltage**.

The presented impact of a representation on search space size could have opposing effects: larger granularity offers more solutions, which are potentially

better. Contrary to that, extending the size of the problem may lead to make it too hard for an evolutionary algorithm to effectively solve in a given number of evaluations.

To compare the results, we use quality indicators commonly used in MOEAs research: (Additive) Epsilon [13], Inverted Generational Distance (IGD) [14], and Spread [15]. The choice of these methods is motivated by their different objectives: Epsilon is the measure of the convergence, while Spread measures the distribution of the solutions. IGD is a metric, which combines both of these components. All of these quality indicators have the minimal values for the best compared methods. Independently to the quality indicators, the number of solutions is taken into account as a metric.

5 Experimental Results

Experiments were conducted using two instance types based on matrix operations applications: Laplace transformation (Laplace) and Gaussian elimination (gaussEl). These application types are commonly used in the high performance computing as well as in benchmarks for scheduling algorithms. The size of an instance depends on the input matrix size and ranges from 25 to 625 for Laplace (21 different sizes), and from 25 to 403 for gaussEl (22 different sizes). The instances can be characterized also by the number of processors, which varies between 8 and 128. Communication to Computation Ratio (CCR) is defined as the ratio of average communication cost of edges to average computation cost of tasks. The tested CCR are 0.1, 0.5, 1, 5, 10. For each combination of number of tasks, number of processors, and CCR an instance was generated, resulting in 525 Laplace instances and 550 gaussEl instances.

Table 1: DVFS characteristics of processors used for simulations.

Level	Type 1		Type 2		Type 3		Type 4		Type 5	
	V_k	R_s (%)	V_k	R_s (%)	V_k	R_s (%)	V_k	R_s (%)	V_k	R_s (%)
0	1.75	100	1.50	100	2.20	100	1.95	100	1.60	100
1	1.40	80	1.40	90	1.90	85	1.60	90	1.30	85
2	1.20	60	1.30	80	1.60	65	1.30	60	1.20	60
3	0.90	40	1.20	70	1.30	50	0.90	40	0.70	40
4			1.10	60	0.90	35	0.60	20		
5			1.00	50						
6			0.90	40						

The machine environment is heterogeneous in two aspects: the generation of execution time, which is done independently for each task and machine, and the DVFS types allocation by round-robin rule to each processor using 5 sets of DVFS characteristics presented in Table 1. Each instance is run independently 50 times for Two-voltage and Multi-voltage approaches with 25,000 evaluations limit, with the crossover probability 0.9 and mutation probability $1/t$. The results of all simulations for an instance are aggregated to produce an approximated true Pareto front, because of the lack of true Pareto fronts for the tackled problem.

Such fronts are used to calculate quality indicators. The problem specifics are implemented as an greenMetal[1] extension to jMetal framework [16] which provides NSGAII algorithm and quality indicators calculation utilities. Due to the size of computational load, an HPC facility of University of Luxembourg[2] was used to speed-up the simulation phase.

The first analysis covers the average number of solutions returned by the algorithm. Figure 3 presents the aggregated values for number of processors, number of tasks, and CCR. For both instance types and for two representations the behavior is similar: the number of solutions decreases for the bigger and harder instances. The main difference is that Two-voltage approach returns on average more solutions than Multi-voltage. Another observation is that the number of processors (Figures 3a and 3d) and CCR (Figures 3c and 3f) have the highest impact on the number of solutions, which tends to converge for number of tasks larger than 150 (Figures 3b and 3e). Reassuming, analysis of number of solutions favors the Two-voltage approach.

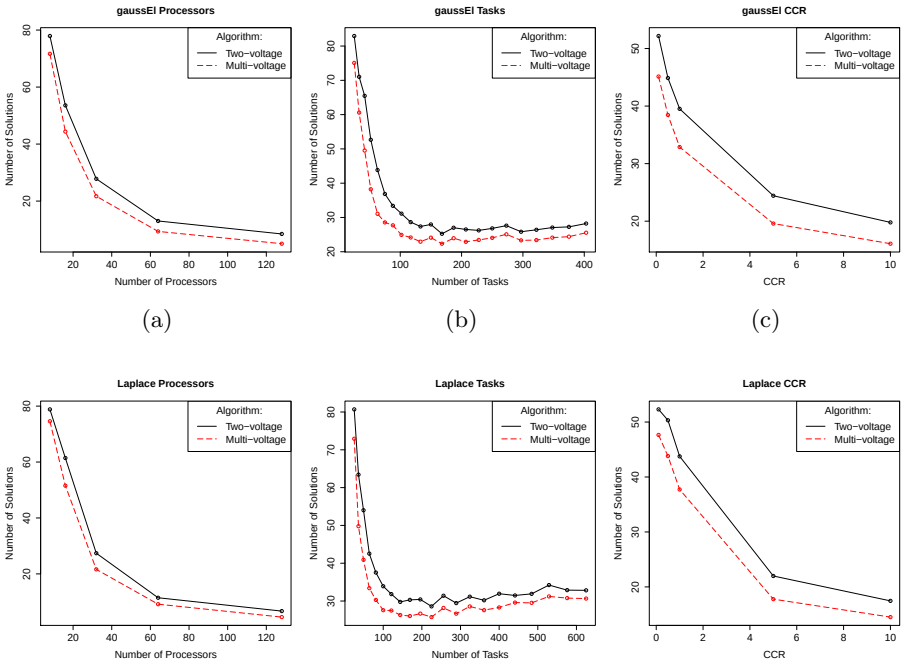


Fig. 3: Number of solutions aggregated by number of processors, number of tasks and CCR.

[1] http://greenmetal.gforge.uni.lu
[2] https://hpc.uni.lu/tiki-index.php

The Friedman and Wilcoxon statistical tests are the selected methods of comparison for the quality indicators mentioned in Section 4, which are calculated separately for each instance. Friedman test performs aggregation of rankings. When two approaches are compared, the Friedman tests values can range between 1 and 2, with the sum of values always being 3. A test result in which one method is ranked with 2 and other with 1 means that one of the methods (ranked with value 2) always outperforms the other method. If both methods have ranks equal to 1.5, it means that they are ranked as the best equally often.

Results of Friedman test for mean and median of the quality indicators are presented in Table 2. In all cases, the *p-value* was lower than $2 \cdot 10^{-10}$. The returned values are similar for mean and median of the independent runs for each instance, what proves that the two representations are relatively stable in comparison to each other. Multi-voltage approach has Friedman test values between 1.79 and 1.86, which gives strong statistical proof that it outperforms Two-voltage approach in terms of the quality of solutions.

Table 2: Friedman tests results.

Instance	Quality indicator	Mean		Median	
		Two	Multi	Two	Multi
gaussEl	Epsilon	1.19	1.82	1.21	1.79
	IGD	1.16	1.84	1.15	1.85
	Spread	1.12	1.88	1.15	1.85
Laplace	Epsilon	1.14	1.86	1.17	1.83
	IGD	1.14	1.86	1.17	1.83
	Spread	1.17	1.83	1.18	1.82

The last statistics performed are aggregations of quality indicators. Wilcoxon signed-rank paired test was used to assure statistical significance of that comparison. The results of aggregation for instance type are presented in Table 3.

Table 3: Values of quality indicators aggregated by instance type.

Aggr. type	Epsilon		IGD		Spread	
Instance type	Two	Multi	Two	Multi	Two	Multi
Laplace	3113	2809	59.32	54.98	1.03	1.00
gaussEl	2778	2598	66.17	61.82	1.05	1.01

In Table 3 and the following tables presenting aggregated quality indicators results, the cells with statistically significantly better (Wilcoxon test *p-value* $<$ 0.5) results are grey colored. The aggregations of results for CCR (Table 4) and for number of processors (Table 5) present the superiority of Multi-voltage representation.

The table for number of tasks is presented only for Laplace instance type (Table 6) due to space limitations and the similarity of results for the gaussEl instances. We can conclude that aggregation results confirm the results of Friedman tests, and underline better quality of Multi-voltage approach.

The experimental results reveals two interesting trends: while Two-voltage representation returns on average more solutions in single run, Multi-voltage approach returns the results of better quality. The outperformance in quality of the solutions is consistent with the MVFS-DVFS heuristic results for a related problem [8].

Table 4: Values of quality indicators aggregated by CCR.

Aggregation type	Laplace						gaussEl					
	Epsilon		IGD		Spread		Epsilon		IGD		Spread	
CCR	Two	Multi	Two	Multi	Two	Multi	Two	Multi	Two	Multi	Two	Multi
0.1	1186	788	2.38	2.67	1.03	0.99	815	529	1.22	1.22	1.07	1.00
0.5	1326	930	10.45	11.96	1.05	1.00	942	677	13.91	13.41	1.08	1.02
1	1451	1050	56.64	47.87	1.06	1.01	1174	955	26.69	26.69	1.09	1.02
5	3867	3683	87.87	83.21	1.02	1.00	3864	3799	249.74	230.74	1.02	1.00
10	7733	7595	160.88	149.29	1.00	1.00	7096	7029	58.27	54.55	1.01	0.99

Table 5: Values of quality indicators aggregated by number of processors.

Aggregation type	Laplace						gaussEl					
	Epsilon		IGD		Spread		Epsilon		IGD		Spread	
Processors	Two	Multi	Two	Multi	Two	Multi	Two	Multi	Two	Multi	Two	Multi
8	1131	996	0.74	0.68	1.04	1.01	800	730	0.49	0.46	1.12	1.05
16	1915	1698	10.78	10.16	1.10	1.03	1305	1208	6.48	6.15	1.12	1.03
32	2347	2051	103.45	94.16	1.02	0.98	1801	1618	21.05	19.32	1.02	0.98
64	3760	3370	41.03	39.00	0.99	0.98	3202	2913	53.04	51.59	0.99	0.98
128	6411	5931	163.37	152.27	0.99	0.99	6782	6520	365.60	338.99	0.99	0.99

Table 6: Laplace instances: aggregated values of quality indicators.

Aggr. type	Epsilon		IGD		Spread		Aggr. type	Epsilon		IGD		Spread	
Tasks	Two	Multi	Two	Multi	Two	Multi	Tasks	Two	Multi	Two	Multi	Two	Multi
25	2262	2141	461.35	410.43	1.13	1.06	256	3082	2597	83.18	78.85	1.02	0.99
36	2083	2009	7.92	7.98	1.09	1.05	289	3140	2930	33.62	33.03	1.02	0.99
49	2359	2237	102.51	100.79	1.08	1.04	324	3601	3157	92.73	89.60	1.01	0.98
64	1916	1749	40.77	43.65	1.07	1.03	361	3828	3413	9.28	8.79	1.01	0.98
81	2150	2061	180.38	160.43	1.06	1.03	400	3810	3390	12.58	12.17	1.02	0.99
100	2838	2676	28.58	28.36	1.06	1.01	441	3882	3519	17.97	17.52	0.99	0.98
121	2571	2276	12.59	11.99	1.06	1.01	484	4066	3622	14.17	13.48	1.00	0.98
144	2965	2701	22.90	22.39	1.04	1.01	529	3789	3375	5.87	5.27	0.99	0.97
169	2893	2629	95.57	82.30	1.04	1.00	576	4067	3689	10.82	10.33	0.98	0.97
196	2926	2528	12.05	11.38	1.03	0.99	625	4202	3821	37.46	31.42	0.98	0.97
225	2936	2476	78.75	77.56	1.03	0.99							

6 Conclusions and Future Work

We showed that the Two-voltage representation provides a Pareto front with more discriminated solutions but of lower quality than the results returned by the Multi-voltage approach. One explanation of the lower number of discriminated solutions in the case of Multi-level voltage can be related to the fact the increase of the search space requires more exploration effort. However due to the fact that a finer tuning is allowed, better quality solutions are faster reached than in the Two-voltage representation. One could imagine that the fact of adding voltage choices corresponds to progressively move from a discrete part of the problem to a continuous one and potentially convex one. The future work directions which we plan to follow are exploration of the problem specifics, which are visible in the aggregated results, as well as identifying convergence properties of algorithms and studying the application of MOEAs of other than NSGAII for this problem. Another open research question is developing the model to cover more aspects of energy efficiency, for example thermal aspect of tasks distribution.

Acknowledgements

This work is supported by the National Research Fund (FNR) of Luxembourg through project Green-IT no. C09/IS/05. Experiments presented in this paper were carried out using the HPC facility of the University of Luxembourg.

References

1. DatacenterDynamics: Datacenterdynamics research report. http://www.datacenterdynamics.com/research Consulted online 2012.
2. Benini, L., Bogliolo, A., De Micheli, G.: Readings in hardware/software co-design. Kluwer Academic Publishers, Norwell, MA, USA (2002) 231–248
3. Herbert, S., Marculescu, D.: Variation-aware dynamic voltage/frequency scaling. In: High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on. (feb. 2009) 301 –312
4. Lee, Y.C., Zomaya, A.Y.: Energy conscious scheduling for distributed computing systems under different operating conditions. IEEE T Parall Distr **22**(8) (aug. 2011) 1374 –1381
5. Pruhs, K., van Stee, R., Uthaisombut, P.: Speed scaling of tasks with precedence constraints. Theor. Comp. Sys. **43** (March 2008) 67–80
6. Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y.C., Talbi, E.G., Zomaya, A.Y., Tuyttens, D.: A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. J Parallel Distr Com **71** (November 2011) 1497–1508
7. Pecero, J.E., Bouvry, P., Huacuja, H.J.F., Khan, S.U.: A multi-objective grasp algorithm for joint optimization of energy consumption and schedule length of precedence-constrained applications. In: DASC, IEEE (2011) 510–517
8. Rizvandi, N.B., Taheri, J., Zomaya, A.Y.: Some observations on optimal frequency selection in dvfs-based energy consumption minimization. J Parallel Distr Com **71** (August 2011) 1154–1164
9. Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
10. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE T Evolut Comput **6**(2) (2002) 182–197
11. Sinnen, O.: Task Scheduling for Parallel Systems. John Wiley & Sons, Hoboken, NJ, USA (2007)
12. Guzek, M., Pecero, J.E., Dorronsoro, B., Bouvry, P., Khan, S.U.: A cellular genetic algorithm for scheduling applications and energy-aware communication optimization. In: HPCS'10, Caen, France (2010) 241–248
13. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (February 2006)
14. Van Veldhuizen, D.A.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. PhD thesis, Wright Patterson AFB, OH, USA (1999) Adviser-Lamont, Gary B.
15. Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., Beham, A.: Abyss: Adapting scatter search to multiobjective optimization. IEEE T Evolut Comput **12**(4) (2008) 439–457
16. Durillo, J., Nebro, A., Alba, E.: The jMetal framework for multi-objective optimization: Design and architecture. In: CEC 2010, Barcelona, Spain (July 2010) 4138–4325