Simultaneous Optimization of Performance, Energy and Temperature for DAG Scheduling in Multi-Core Processors

Hafiz Fahad Sheikh, Ishfaq Ahmad
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, USA
hafizfahad.sheikh@mavs.uta.edu, iahmad@uta.edu

***Abstract*—This paper addresses the joint optimization of performance, energy, and temperature, termed as PET - optimization. This multi-objective PET-optimization is achieved in scheduling DAGs on multi-core systems. Our technique is based on multi-objective evolutionary algorithm (MOEA) for finding Pareto optimal solutions using scheduling and voltage selection. These solutions are not necessarily scalar values but can be in a vector form. We developed a Strength Pareto Evolutionary Algorithm [2] (SPEA) based solution which is inherently superior to several other MOEA methods. The proposed algorithm obtains the Pareto vectors (or fronts) efficiently. The work is novel and original in the sense that no previous such optimization work has been reported to our knowledge for the PET-optimization scheduling problem. The strength of the proposed algorithm is that it achieves diverse range of energy and thermal improvements while staying close to the performance-optimal point to ensure efficient trade-off solutions. The proposed approach consists of two-steps. In the first step, Pareto fronts are generated. In the second step, one most optimal solution is selected. Simulation results on several benchmark task graph applications demonstrate that efficient solutions can be selected using the proposed selection method in polynomial time.**

Keywords: dynamic thermal management, multi-objective evolutionary algorithms, multi-core systems, task scheduling.

I. INTRODUCTION

The expeditious increase in the number of cores on a chip demands novel and effective software tools for managing the power dissipation by multi-core systems. Higher power dissipation not only requires higher cooling costs but also results in multiple thermal issues. An unchecked temperature can impact the performance [10] as well as reliability and life span of the modern multi-core systems. For instance, the expected life time of a system can be reduced to half due to a 10^{o}C increase in the operating temperature [16]. Similarly, it has also been observed that an increase in the magnitude and frequency of thermal cycles has a deteriorating effect on the reliability of the system [11]. Fortunately, most of the current multi-core systems support dynamic voltage/frequency scaling (DVFS) and low power sleep states. Using these control mechanisms various task scheduling, task migration and core consolidation policies can be designed to efficiently manage the temperature and energy consumption. Although, considerable research has been conducted for thermal and energy-aware task allocation and task scheduling, most of the existing algorithms are generally limited to the optimization of either the temperature or energy or performance, subject to a constraint. A combined approach for scheduling tasks on multi-core processors that considers performance, energy and temperature simultaneously remains a rather unexplored area. In this paper we address the joint optimization of performance, energy, and temperature (*PET quantities*). The target is to maximize performance (minimize makespan) and at the same time to minimize the energy consumption and the peak temperature while allocating tasks on a multi-core system. This multi-objective prospect adds to the complexity of an already NP-hard task scheduling problem.

We present a two-step solution to the multi-objective optimization (MOO) problem for allocating tasks represented as task graphs on a set of available cores and at the same time selecting a voltage level for the execution of each task. In the first step, we develop a novel approach for obtaining Pareto optimal solutions leveraging a multi-objective evolutionary algorithm (MOEA). Specifically, we used SPEA-II [2] to generate the Pareto optimal solutions for the MOO problem. Our results indicate that SPEA-II based approach can generate Pareto fronts comprising of multiple schedules for the given task graph to facilitate efficient trade-offs among the *PET quantities*. As second step, we present a weighted-sum based solution-selection heuristic which can quickly select the required solution automatically or with a user's preference from a given set of Pareto optimal solutions. One strength of the proposed approach is that it provides multiple solution points for simultaneously optimizing performance, energy, and temperature, rather than merely adjusting a given schedule for the imposed energy or thermal constraints. Each of these points leads to a complete schedule for the given set of tasks, enabling a decision maker to achieve efficient trade-offs between the *PET quantities*.

The rest of the paper is organized as follows: Section II covers the related work. Section III presents the multi-objective optimization (MOO) problem under consideration. Section IV provides the details of the proposed solution to the problem and Section V highlights the experimental setup. Section VI explains the results of simulation while Section VII concludes the paper along with some directions for future work.

II. RELATED WORK

Several research efforts use evolutionary algorithms to solve task scheduling and allocation problems for different settings. However, none of these addresses the issue of optimizing performance along with energy consumption and thermal profile. An evolutionary approach to schedule batch tasks on a distributed system is presented in [3]. The approach targets

both the static and dynamic task allocations to address the dynamic condition of distributed system at the time of task scheduling. To address the multidimensional QoS issues in grid environment, [4] presents an NSGA-II [1] based scheduling scheme whereas [5] aims to optimize the throughput for computational grids. An evolutionary scheme to schedule DAGs on multiprocessors for minimizing the task completion time has been proposed in [6]. A fast and efficient task scheduling and voltage selection approach called Evolutionary Relative Slack Distribution Voltage Scheduling (ERSD-VS) has been introduced in [9].

Research efforts [12], [13], [14], [15] aim to jointly address the issue of performance degradation and thermal management mostly using heuristic approaches. Pre-built look-up tables representing the thermal effects due to power activation at different locations of the chip have been used in [12] to allocate tasks to different cores. The scheme targets to improve the peak temperature as well as guarantee the thermal limit while at the same time decreases the rejection ratios under thermally constrained CMPs. [13] attempts to resolve issues related to the application of conventional DVFS schemes using hardware-software co-design. An operating system level heuristic of scheduling hot-job before a cool-job to minimize the number of DTMs has been proposed in [14]. This approach called ThreshHot, schedules the hottest job which does not violate the thermal limit from a given set of tasks while making scheduling decisions. A convex optimization problem for allocating frequencies to different cores under different workloads for minimizing the power consumption to meet the thermal thresholds and to reduce the hotspots has been solved in [15].

Some of the research efforts report the improvement in either performance and energy or performance and temperature [17]. However, these improvements are only a side effect of an efficient DTM scheme and not due to the joint optimization. In contrast, our work solves the task assignment and voltage selection problem for simultaneously optimizing performance, energy, and temperature to attain efficient trade-offs among the *PET quantities*.

III. Problem Formulation

We consider the problem of minimizing makespan, energy consumption, and peak temperature while scheduling a set of tasks on a given multi-core system with M cores. Our workload consists of applications, each represented as a directed acyclic graph (DAG). A DAG(N,E) comprises of N nodes and E edges. The weight associated with each node represents the number of cycles required to complete that task, whereas weight on each directed edge represents the number of cycles required to transfer data after the completion of the parent task to its successor task. A path in the DAG with longest length (sum of the weights of nodes and edges along the path) from any entry node to an exit node is called the Critical Path (CP). The tasks or nodes on CP are known as Critical Path Nodes or CPNs. The nodes which have a successor on CP are called Inbound Nodes or (IBNs). All other tasks are considered as Outbound Nodes (OBNs). While allocating tasks to the given set of cores we prioritize tasks according to their type, that is CPNs are considered first, then IBNs and finally OBNs. However, while constructing this priority list it is ensured that parent nodes are added to the list before each node itself. The resulting list is also known as CPN-DS (CPN Dominant Sequence [24]).

Given the priority of each task as obtained using CPN-DS, we aim to find the task-core mappings and the voltage level for the execution of each task to minimize the required objectives. If there are N tasks, M cores and K voltage levels, we define the *PET*-optimization problem as:

$$Minimize \quad \max_{1\le j\le M} s_N^j \tag{1}$$

$$Minimize \quad \sum_{i=1}^{N} p_i\, et_i \tag{2}$$

$$Minimize \quad \max_{1\le i\le N} \max_{1\le j\le M} T_i^j \tag{3}$$

Equation (1) refers to the minimization of makespan where $s^j{}_N$ represents the finish time of jth core after the completion of Nth task. In (2), et_i is the execution time and p_i is the power consumed by ith task. T^j_i in (3) represents the peak temperature of the jth core during the execution of ith task. We have not imposed any constraint related to performance, energy, or temperature in the problem above. By doing so, we focus on obtaining a general Pareto-front for the problem. The constraints can be imposed in solution selection phase to filter out only those schedules which satisfy the given requirements.

IV. Proposed Solution

In this section, we first explain the Pareto front generation followed by the method developed for selecting a solution from these Pareto fronts.

A. Pareto Front generation using SPEA-II:

Every possible solution to the MOO problem in (1)-(3) represents a task-core mapping and the voltage selection for each task in decision vector space and a corresponding value of makespan, energy consumption, and peak temperature in objective space. These objectives are in conflict with each other. For example, minimizing peak temperature requires lower power dissipation which can potentially increase makespan and vice versa. In the presence of multiple conflicting objectives, the concept of optimality is transformed to Pareto optimality, as any solution point has to be gauged along multiple dimensions. Thus, in general, given two solutions x_1 and x_2 for a given multi-objective optimization problem with m objectives, we can define the dominance relationship $(\prec)$ between the two solution points as:

$$x_1 \succ x_2 \; if \; \exists k \,|\, f_k(x_1) < f_k(x_2) \wedge f_j(x_1) \le f_j(x_2), \forall\, 1 \le j,k \le m \tag{4}$$

In the above definition, we assume that each objective has to be minimized and $F(x) = \{f_1, f_2, ..., f_m\}$ represents the set of all objective functions. Now, given a set of solutions $X = \{x_1, x_2, ..., x_n\}$, we can construct a set of Pareto optimal solutions as $P = \{x_i \,|\, \nexists x_j \succ x_i, \forall 1 \le i, j \le n\}$.

To generate these Pareto optimal solutions we have used Strength Pareto Evolutionary Algorithm (SPEA) [7]. SPEA-II [2] which is an improved version of this approach has been shown to yield better Pareto fronts when compared with other

approaches like PESA [21], VEGA [18] and NSGA-II [1] (for high dimension problems). Also, the ability of SPEA-II to obtain Pareto fronts closer to the global Pareto fronts is well established. The motivation behind using an evolutionary approach is to obtain the Pareto fronts with low computational complexity. Overall, SPEA-II follows the generic evolutionary process which includes population initialization, reproduction (crossover and mutation), population selection, truncation etc. Here, we will only briefly discuss the details of each of these steps but more details can be found in [1], [2], [7], [8], [18]. Nevertheless, we will highlight the key aspects of SPEA-II in terms of fitness assignment, diversity preservation and population truncation as they play a key role in guiding a set of population members towards the global Pareto front.

1) Initial population

Initial population is generated using uniform distributions defined over the range of each variable. For the MOO problem in (1)-(3), the range of variables is governed by the total number of cores (M) and the number of available voltage levels (K). For a DAG of size n each population member (solution) is a string of size $2n$. Indexes from 1 to n correspond to the task allocation decisions where as voltage selection decisions are encoded with variables at indexes from n+1 to $2n$. K members of the initial population are initialized with the allocations as obtained through DCP [24] with every task running at the same *kth* voltage level.

2) Elitism

SPEA-II uses two sets of populations, the archive or external population and the current or offspring population. The external population holds the best solution found so far and thus adds elitism to the evolution. Non-dominant solutions after each generation are copied to the external generation. However, if the size of external population exceeds a threshold then a truncation operator is used to maintain the required number of the members in the external population. In SPEA-II a clustering based technique is used. The clustering technique removes the individuals from the archive based on the minimum distance to another solution in the objective space.

3) Fitness Assignment

For multi-objective optimization the non-domination of any solution is a main indicator of its Pareto optimality. Therefore, SPEA-II incorporates the non-dominance information while assigning fitness to every population member. In the earlier version of SPEA, separate fitness functions were used for the archive generation (G') and the current generation (G) but SPEA-II uses the same fitness assignment function for both types of population members. So while assigning fitness, both the archive and the current population members are considered as opposed to SPEA-I which takes into consideration the solutions in archive only. Now, assuming i and j are the population members ϵ G' U G, then fitness to each member can be assigned as:

$$fitness(i) = Strength(i) + Density(i) \tag{5}$$

The above fitness assignment to a solution i has two parts. *Strength (i)* is based on the domination strength and can be given by:

$$Strength(i) = \sum_{j} DS(j),\ \forall\ j \in G'UG\ \wedge\ j \succ i \tag{6}$$

where

$$DS(j) = \sum_{i'} x_{j,i'},\ \ \forall\, i' \in G'UG \wedge x_{j,i'} = 1 iff\ \ j \succ i' \tag{7}$$

The second component of fitness value represents the density of solution points around the selected population members. The density estimate is based on the *kth n*earest-neighbor approach, where the actual value of the density estimate is equal to the inverse of the Euclidean distance of the *kth*-nearest neighbor in the objective space and can be given by:

$$Density(i) = \frac{1}{d_i^k + 2} \tag{8}$$

where $d_i^k = \|i - k_{NN}\|$ (8a)

In (8a), k_{NN} represents the *kth* nearest neighbor of the population member i in both populations. The time complexity of the proposed fitness assignment strategy is $O(\sigma^2 log\sigma)$, where σ is the total number of solutions in the archive and the current population.

4) Genetic Operations

The members of mating pools are selected by using binary tournament selection. While comparing the two members of the population, the fitness values in conjunction with the density information are used to decide the winner of any round. With regards to genetic operations, we introduced modified cross-over operator for SPEA-II to suit our problem which is formulated as an Integer Programing problem. Specifically, we use a binomial distribution with mean set to the value of parent variable and variance proportional to the difference between the two parents of an offspring. This distribution is then used to obtain the value of the corresponding variable for the offspring member. These modifications to the crossover operator maintain the following important properties as outlined in [19]:

- The possible range of values that a decision variable can take in an offspring during cross-over is proportional to the difference in the values of parent members for the same decision variable.
- The mean of probability distribution for the possible values that an offspring can take is set to the value in the parent member.

This allows the offspring to take values close to that of the parent when the difference among the parents is small and at the same time provides the opportunity for the offspring to differ significantly from parent member, when the difference among them is large.

5) Computational Complexity

The computationally dominating part in SPEA-II is the archive update procedure. Let us assume that there are a combined "σ" population members in the archive and current populations and the dimension of objective space is N_o. Then the average-case time-complexity for updating the population (in each generation) comes out to be $O(\sigma^2 log\ \sigma)$, therefore, the overall average-case time-complexity of SPEA-II can also be considered as $O(\sigma^2 log\ \sigma)$.

TABLE 1
DVFS PARAMETERS

f(MHz)	1600	2000	2200	2400	2600
P(W)	23.61	48.90	72.48	93.12	105.00

TABLE 2
SYSTEM AND EVOLUTION PARAMETERS

Parameter	No. of Cores	Layout	Freq. Switching	Size of Population	No. of Generations	Cross- over prob.	Mutation Prob.
Value	16	Grid 4x4	Independent	30	40	0.8	0.2

B. Solution Selection

Selection of an operating point or solution from the given set of generated Pareto fronts is not straight forward due to the non-dominance relationship that exists among the solutions. The first step in solution selection is to apply different constraints (if present) to obtain the feasible Pareto set ($P_{feasible}$). These constraints can be in the form of thermal limit, energy budget, or a deadline for the completion of the task set. Next, we assume that a preference vector of the form [α ϒ β] is available. This preference vector is used to define the corresponding weight for each objective resulting into a weight vector of the form $W = [\alpha/\lambda \;\; \gamma/\lambda \;\; \beta/\lambda]$ where λ=α+ϒ+β. Taking the minimum value along each objective as reference we rank each solution (x) in the feasible Pareto set as:

$$rank_x = \sum_{m=1}^{N_o} w_m \frac{f_m(x)}{f_m^*}, \;\; \forall x \in P_{feasible}, \;\; w_m \in W \tag{12}$$

Where f^*_m represents the minimum value for the mth objective over the whole Pareto front. N_o represents the total number of objectives (in our case N_o=3). A second scenario exists for solution selection phase where no preference vector is available. For such a case, a weight vector of the form [1 1 1] can be used to rank the solutions. Regardless of the scenario, we select the solution which results in the minimum value of *rank*. In other words the selected solution is given by

$$x^{selected} = \arg\min_{\forall x} \; (rank_x), \;\; \forall x \in P_{feasible} \tag{13}$$

The use of the proposed solution selection scheme under various scenarios will be presented in Section VI.

V. EXPERIMENTAL DETAILS

We have assumed a 16 core system with cores arranged in a grid layout of 4x4. Each core was assumed to be able to switch independently through 5 different frequency/power settings in its active mode. Table 1 presents DVFS parameters while Table 2 lists different system and evolution parameters. Specific details about thermal model and workload are presented below.

A. Thermal Model

For thermal model we have used the steady state temperature based on the average power consumption of a core during an interval which can be given by:

$$T_j = R_{th} P_j + T_A \tag{14}$$

where T_A is the ambient temperature, R_{th} is the thermal resistance of the system and P_j and T_j represent the power consumption and temperature of jth core in each interval. This thermal model is simple and effective but it ignores the lateral effects and the power consumption profile of the neighboring cores. To incorporate the neighbor effect we have modified (14), similar to [20] as:

$$T_j = R_{th} P_j + \sum_{\forall m \in neighbor_j} \gamma R_{th} P_m + T_A \tag{15}$$

where, $neighbor_j$ is the set of cores which are adjacent to the jth core. γ is the coefficient which relates the effect of power consumption of a particular core on the temperature of its neighboring cores.

It is important to note that the corresponding values of performance, energy and temperature resulting due to certain allocation and voltage selection decisions can be derived from any model and therefore, our approach is independent of any underlying model including those that include the effects of leakage power into the thermal model or variations in the task execution profiles.

B. Workload

We considered the task graphs of diverse type of applications that include Fast Fourier Transform (FFT) [25], Gauss Elimination [26], Laplace Equation [26], as well as two application task graphs from Standard Task Graph set (STG) [22] namely Robot Control and Fpppp [23] .

VI. RESULTS

A. Pareto Front generation using MOEAs

Figs. 1-3 present the Pareto fronts obtained for the task graph applications of FFT, Robot Control, and Fpppp. Each point in these Pareto fronts shown in objective space is a complete schedule which corresponds to a task-core mapping and the selected voltage level for each task in the decision vector space. We observe that SPEA-II based approach can generate multiple solutions (schedules) which are well spread along the Pareto front. Figs. 4-6 delineate the performance-energy and performance-temperature trade-off relationships for the corresponding task graphs. It can be noticed that SPEA-II generated a large number of distinct schedules to judiciously trade-off performance for improving energy consumption and peak temperature. Table 3 presents the minimum, maximum, and average, values along with the corresponding range, standard deviation and percentage difference between the maximum and minimum values along each objective. The percentage difference between the maximum and minimum values along performance, energy, and temperature can be as high as 62.43%, 90.81%, and 67.80% respectively. In addition, the average values of percentage differences over all the task graph applications for the *PET quantities* are very close to the peak percentage difference along each objective.

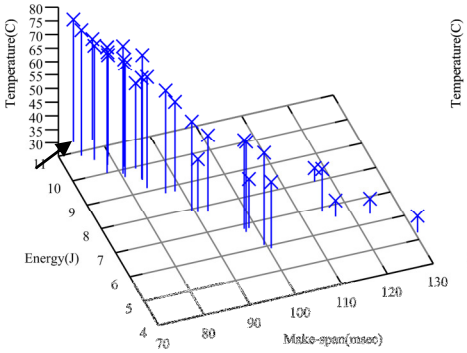

Fig. 1. Pareto front for FFT.

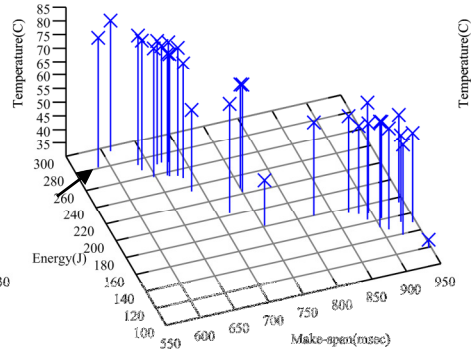

Fig. 2. Pareto front for Robot Control.

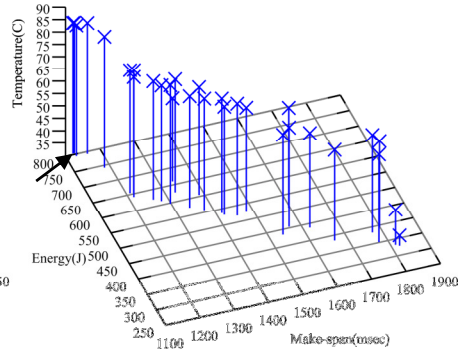

Fig. 3. Pareto front for Fpppp.

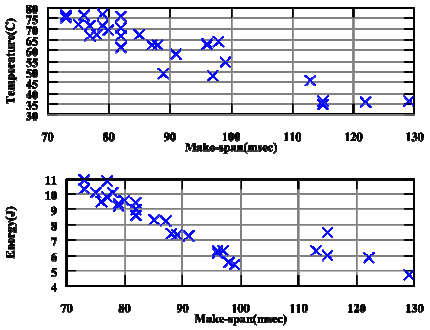

Fig. 4. Trade-off curves for FFT.

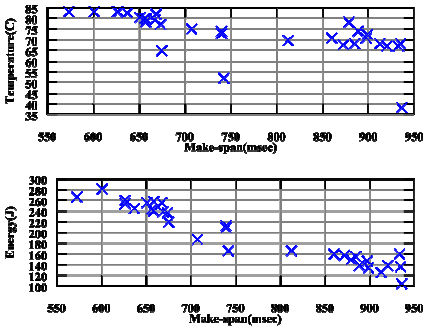

Fig. 5. Trade-off curves for Robot Control.

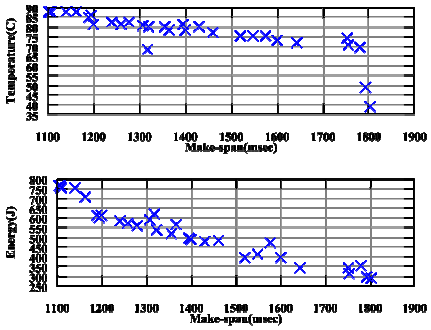

Fig. 6. Trade-off curves for Fpppp.

This indicates that SPEA-II is able to generate efficient Pareto fronts for diverse type of applications/task graphs.

The actual value of performance degradation is very critical for schemes which present solutions to trade-off performance with energy and temperature. Therefore, for reference, we used the performance optimal schedules as generated by DCP (that can generate near optimal schedules [24]) for each task graph application. The corresponding values of the *PET quantities* are shown with arrow on each Pareto front. We notice that Pareto fronts obtained by SPEA-II based approach are always tangentially very close to this performance optimal point; this validates the quality of trade-off solutions as generated by SPEA-II.

B. Solution Selection

Table 4 lists the objective values for the solutions comprising the Pareto front of Robot Control application obtained using SPEA-II approach. We can now impose any form of constraint on the given set of solutions to get the feasible Pareto solutions. As an example we have used a thermal constraint of 72°C. The solutions from 13-30 do not satisfy the imposed constraint and therefore will not be considered during solution selection. As a first scenario we assume that no preference vector is given, for this case we use [1 1 1] as preference vector. For the second scenario we assume that [8 1 1] is the preference vector. Column 4 and 5 in Table 4 present the corresponding values of *rank* for every member and the deviation in the *PET quantities* (as compared to the DCP schedule) for the selected solution under each scenario. Solution#1 is picked up under the first scenario

TABLE 3
DIFFERENT ATTRIBUTES OF PARETO FRONTS ALONG EACH OBJECTIVE

Qt.	Task Graph	Min	Mean	Max	Range	Std. Dev.	%Diff
Performance	Fft	73	90	129	56	44	62.43%
	Gauss	328	396	494	166	136	41.91%
	Laplace	760	934	1264	504	121	53.98%
	Robot	573	764	936	363	170	47.49%
	Fpppp	1103	1403	1802	699	309	49.81%
	Average						**51.12%**
Energy	Fft	4778	8138	10899	6122	10920	75.22%
	Gauss	29774	49973	68605	38831	28560	77.70%
	Laplace	66767	121043	163405	96638	52011	79.84%
	Robot	105157	196700	280619	175462	62972	89.20%
	Fpppp	289023	522977	763953	474931	157682	90.81%
	Average						**82.56%**
Temperature	Fft	35	62	77	42	9	67.80%
	Gauss	37	65	78	41	13	62.19%
	Laplace	37	62	78	41	10	66.09%
	Robot	38	73	83	45	8	62.03%
	Fpppp	39	77	88	49	9	63.20%
	Average						**64.26%**

which represents an improvement in energy consumption by 61% and reduction in peak temperature by 54% and yields a makespan that is 1.64 times than that of the performance optimal schedule. Solution#3 is picked up for the second scenario achieving 22% and 17% reduction in energy consumption and peak temperature respectively while degrading the performance by 18%. It should be noted that during the selection phase, only a single term needs to be evaluated for each solution, therefore for a set of population of size ω, the total complexity of solution selection phase is $O(\omega)$.

TABLE 4
SOLUTION SELECTION

Sol #	P(msec)	E(mJ)	T(°C)	$rank_x^1$	$rank_x^2$
1	9.36E+02	1.05E+05	3.80E+01	**3.63**	1.51
2	7.42E+02	1.67E+05	5.19E+01	4.25	1.33
3	6.75E+02	2.20E+05	6.49E+01	4.98	**1.32**
4	9.33E+02	1.61E+05	6.70E+01	4.92	1.63
5	9.20E+02	1.39E+05	6.71E+01	4.69	1.59
6	8.72E+02	1.57E+05	6.75E+01	4.79	1.54
7	9.35E+02	1.36E+05	6.80E+01	4.71	1.61
8	9.13E+02	1.25E+05	6.82E+01	4.58	1.57
9	8.85E+02	1.56E+05	6.82E+01	4.82	1.56
10	8.13E+02	1.66E+05	6.94E+01	4.82	1.48
11	8.60E+02	1.61E+05	7.06E+01	4.89	1.54
12	8.98E+02	1.47E+05	7.06E+01	4.83	1.58
13	8.99E+02	1.34E+05	7.23E+01	-	-
-	-	-	-	-	-
30	6.01E+02	2.81E+05	8.33E+01	-	-
Min	5.73E+02	1.05E+05	3.80E+01	3.63	1.32
P_{opt}	5.73E+02	2.67E+05	8.28E+01		

Selected under first Scenario
ΔPerf. = 63%
ΔE = -61%
ΔT = -54%

Selected under 2nd Scenario
ΔPerf. = 18%
ΔE = -22%
ΔT = -17%

VII. CONCLUSION

The proposed SPEA-II based approach obtained Pareto fronts with solution points presenting a wide range of values for each objective. The results indicate that the Pareto fronts generated by SPEA-II exhibited a uniform spread and obtained multiple schedules for each task graph to trade-off performance with energy and thermal improvements at a fine grained level. We also presented an algorithm for selecting the best optimal solution from the Pareto fronts obtained using the SPEA-II based approach. We demonstrated that the proposed selection algorithm can pick solutions accurately and quickly (polynomial time) under various scenarios. An important aspect of our proposed technique is that it is independent of any underlying models and assumptions about the architecture; therefore, our approach is robust and applicable to a wide range of systems. An extension to this work is the inclusion of the state of the system in the solution selection phase to switch across different Pareto optimal solutions during the course of execution of a given task set.

ACKNOWLEDGMENT

This work was supported by grants from the National Science Foundation under contract #CCF-0903456, #CCF-0905308, and CRS- 0905196 and SRC under contract #2009-HJ-1978.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on* , vol.6, no.2, pp.182-197, Apr 2002.

[2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Swiss Federal Institute of Technology, Dept. of Elect. Engineering, Tech. Rep. TIK-Report 103, 2001.

[3] Z. Flavia, Z. Daniela, C. Ciprian, "Evolutionary task scheduling in static and dynamic environments," *Computational Cybernetics and Technical Informatics, International Joint Conference on*, pp.619-624, May 2010.

[4] H. SongFa and Z. Ying, "NSGA-II based grid task scheduling with multi-qos constraint," *Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on*, pp.306-308, 2009.

[5] S. Zheng, W. Shu, D. Shangping, "Task scheduling model design using hybrid genetic algorithm," *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on*, vol.3, pp.316-319, 2006.

[6] P. Switalski and F. Seredynski, "Solving Multiprocessor Scheduling Problem with GEO metaheuristic," *Parallel & Distributed Processing, 2009, IEEE International Symposium on*, pp.1-8, 23-29 May 2009.

[7] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength pareto approach," Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 1998.

[8] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, Vol. 2, No. 3, Pages 221-248, 1994.

[9] B. Gorji-Ara, C. Pai, N. Bagherzadeh, M. Reshadi, and D. Jensen, "Fast and efficient voltage scheduling by evolutionary slack distribution," *Design Automation Conference, 2004. Proceedings of the ASP-DAC 2004. Asia and South Pacific*, pp. 659- 662, 27-30 Jan. 2004.

[10] R. Viswananth, V. Wakharkar, A. Watwe, and V. Lebonheur, "Thermal Performance Challenges from Silicon to Systems," *Intel Technol. J., Q3,* vol. 23, p. 16, 2000.

[11] A. H. Ajami, K. Banerjee, and M. Pedram, "Modeling and Analysis of Nonuniform Substrate Temperature Effects on Global Ulsi Interconnects," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 849-861, Jun. 2005.

[12] J. Cui and D. L. Maskell, "Dynamic Thermal-Aware Scheduling on Chip Multiprocessor for Soft Real-Time Systems," *Proceedings of the 19th ACM Great Lakes Symposium on VLSI, GLSVLSI '09*, May 2009.

[13] O. Khan and S. Kundu, "Hardware/Software Co-design Architecture for Thermal Management of Chip Multiprocessors," *Conference & Exhibition on Design, Automation & Test in Europe*, April 2009.

[14] J. Yang, X. Zhou, M. Chrobak, Y. Zhang, L. Jin, "Dynamic Thermal Management through Task Scheduling," *IEEE International Symposium on Performance Analysis of Systems and software, ISPASS 2008*, pp.191-201, 20-22 April 2008.

[15] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. De Micheli, "Temperature Control of High-Performance Multi-Core Platforms using Convex Optimization," *Proceedings of the Conference on Design, Automation and Test in Europe, ACM*, pp. 110-115.

[16] JEDEC Solid State Technology Association, Arlington, VA, "Failure mechanisms and models for semiconductor devices," *JEDEC publication JEP122C*, 2006. [Online]. Available: http://www.jedec.org

[17] T. Ebi, M. Faruque, and J. Henkel, "TAPE: Thermal-aware Agent-based Power Economy Multi/many-Core Architectures," *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, ICCAD 2009*, pp.302-309, November 2009.

[18] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," *Proceedings of the 1st International Conference on Genetic Algorithms*, p.93-100, July 01, 1985.

[19] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

[20] Z. Wang and S. Ranka, "A Simple Thermal Model for MPSoC and its Application to Slack Allocation," *Proceeding of IEEE International Parallel & Distributed Processing Symposium*, 2010.

[21] D. W. Corne, J. D. Knowles, and M. J. Oates , "The pareto envelope-based selection algorithm for multiobjective optimization," In *Parallel Problem Solving from Nature – PPSN VI, Berlin*, pp. 839–848. Springer.

[22] Standard Task Graph Set, "STG", [Online]. Available: http://www.kasahara.elec.waseda.ac.jp/schedule/

[23] Standard Performance Evaluation Corporation, "SPEC", [Online]. http://www.spec.org/cpu95/CFP95/145.fpppp/.

[24] Y-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Comput. Surv.* vol 31, no. 4, pp. 406-471, 1999.

[25] I. Ahmad, Y.-K. Kwok, M.-Y. WU, and W. Shu, "CASCH: A Tool for Computer-Aided Scheduling," IEEE Concurrency, vol. 8, no. 4, pp. 21-33, October 2000.

[26] M.-Y. Wu, D. Gajski, "Hypertool: A Programming Aid for Message-Passing Systems," IEEE Transactions on Parallel and Distributed Systems, vol.1, no. 3, pp. 330-343, July 1990.