Multiobjective Scheduling of Green-Powered Datacenters Considering QoS and Budget Objectives

Santiago Iturriaga
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
Email: siturria@fing.edu.uy

Sergio Nesmachnow
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
Email: sergion@fing.edu.uy

***Abstract*—This article presents a multiobjective approach for scheduling green-powered datacenters. We consider a bag of independent deadline-constrained tasks to be executed in a datacenter partially powered by green energy where machines and be powered on/off. The problem consists in scheduling machine state, task execution, and cooling devices to follow an energy consumption profile while simultaneously minimizing the operational budget and the QoS degradation, subject to maintaining the datacenter temperature below its maximum operational threshold. We propose an Evolutionary Algorithm empowered by a Local Search for tackling this problem. Preliminary results show promising budget reductions when compared to a greedy scheduling approach.**

I. Introduction

The energy consumption of datacenters has became a major concern in the last decade, accounting for approximately 1.5% of the world total energy usage [1]. Datacenter providers, and governments alike, are searching for greener energy-aware solutions to reduce brown energy consumption for economic and environmental reasons. An approach for tackling this problem consists in dynamically controlling the operation of a datacenter according to some energy-related conditions. This allows to reduce the budget and the brown energy consumption of the datacenter by scheduling the bulk of its workload to when there is green energy available or when brown energy is cheaper. The use of green energy sources allows the datacenter to reduce its brown energy consumption, but because of their unreliable nature, most datacenters cannot commit to green energy sources alone.

We present a multiobjective approach for controlling energy consumption and budget in datacenters, while also considering the Quality of Service (QoS) provided to the users. The proposed approach controls the datacenter operation by scheduling its workload, powering its servers on and off, and adjusting the datacenter's temperature by controlling its cooling devices. This allows the datacenter to minimize the amount of power consumed subject to given power profile, while at the same time, minimize the required budget and the QoS degradation.

The workload of the datacenter is represented by a set of independent tasks with deadlines submitted by its users. It is important to consider a QoS-related objective since reducing energy consumption in a computing system most certainly will affect the QoS provided to the users. In our approach, the QoS degradation of the system is measured by the missed deadlines of the tasks in the workload.

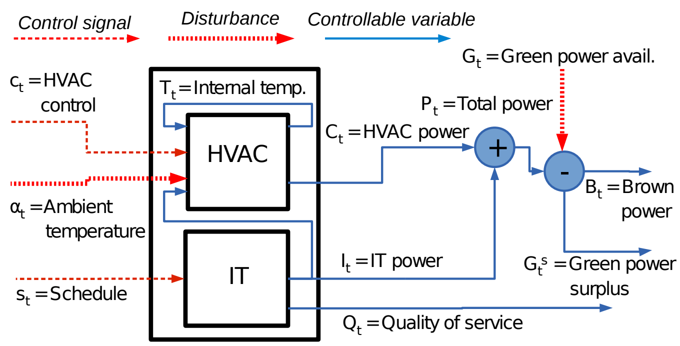

Fig. 1. Diagram of the inputs and outputs of the datacenter.

A similar approach was presented in [1], [2]. [2] proposes a datacenter powered only by green energy, and considers a power generation profile, QoS and temperature as optimization objectives. They use a single objective approach and introduce a greedy heuristic for solving the problem. [1] extends [2] by presenting a MultiObjective Evolutionary Algorithm (MOEA) for solving the problem proposed in [2]. This enhances the previous greedy scheduling approach by improving its accuracy and producing not one, but a set of trade-off solutions. In this work, we further extend [1], [2] by considering brown energy as a viable energy source and by considering budget as an optimization objective. Furthermore, we propose an improved scheduling algorithm by hybridizing the previously proposed MOEA with a Local Search (LS) operator for exploiting the most promising solutions.

II. Datacenter Model

The datacenter model used in this work is based on the model proposed in [1]. Figure 1 shows the schema of the model.

We consider two power consuming components, the *Heating-Ventilation-Air Conditioning* (*HVAC*) and the computing infrastructure (*IT*). The red, thin, dotted arrows represent *control signals*; the red, fat, dotted arrows represent external *disturbances*; and the blue, solid arrows represent *controllable variables*.

Control signals are variables which can be changed by the scheduling algorithm to modify the state of the datacenter: the HVAC control (c_t) regulates the operation of the HVAC system, while the Schedule (s_t) schedules which servers are on, which are off, and where and when to execute each task.

Disturbances are variables which are external and are outside of our control: ambient temperature (α_t) represents the temperature outside the datacenter and green power available (G_t) represents the green power generated by the renewable energy source.

Controllable variables are output variables whose values are determined by the control signals and the disturbances: the QoS variable (Q_t) depends on the deadlines missed by the schedule, the internal temperature (T_t) is the thermostats reading inside the datacenter, the cooling power (C_t) is the sum of AC power and free cooling fan power, the IT power (I_t) is the power consumed by the servers, switches and all IT equipment in the datacenter, the green power surplus (G_t^s) determines the excedent of renewable energy not used by the datacenter, and brown power (B_t) represents the amount of energy consumed from the power grid.

III. Scheduling Problem

The goal of this work is to schedule the *control signal* variables during a period of time so that the total power consumption (P_t) does not exceed a given power profile (R_t), while the brown energy budget and QoS degradation are minimized, subject to maintaining the internal datacenter temperature (T_t) below its maximum operative value.

The datacenter executes N tasks in a simulation period of K timesteps. Each task i requests to be executed before a deadline $D(i)$. $FT(i)$ represents the finishing time of task i according to the considered schedule. The brown energy cost value, M_t^b, defines the monetary value of a energy unit on each timestep.

Formally, we want to simultaneously *minimize*:

$$z_p = \sum_{t=1}^{K} \begin{cases} (P_t - R_t)/\max(R_t) & \text{if } P_t > R_t \\ 0 & \text{if } P_t \leq R_t \end{cases} \tag{1a}$$

$$z_b = \sum_{t=1}^{K} B_t \times M_t^b \tag{1b}$$

$$z_q = \sum_{i=1}^{N} \begin{cases} FT(i) - D(i) & \text{if } FT(i) > D(i) \\ 0 & \text{if } FT(i) \leq D(i) \end{cases} \tag{1c}$$

Objective (1a) specifies the power consumption of the system should not be above the reference power profile. Objective (1b) is the total monetary cost of the energy consumption of the system. Objective (1c) represents the total time of the deadline violations.

The datacenter in our model is comprised of two subsystems: HVAC (cooling) and IT. IT power I_t is calculated as $I_t = S_t^{max} + S_t^{idle} + S_t^{sleep}$ where S_t^{max}, S_t^{idle} and S_t^{sleep} are the total power consumption of all servers that are executing, idle and sleep at time t respectively. We consider two cooling modes: air conditioning and free cooling. In air conditioning mode, we use a conventional direct expansion Computer Room Air Conditioner (CRAC), which can take the values on/off. In free cooling mode, the CRAC is turned off and outside air is blown into the datacenter by a fan. Formally:

$$C_t = \begin{cases} \mathit{CompressorPWR} & \text{if in AC mode, compressor ON} \\ 0 & \text{if in AC mode, compressor OFF} \\ \mathit{FanPWR} & \text{if in free cooling mode} \end{cases}$$

The value of C_t and the cooling mode directly affect the temperature T_t in the datacenter. As shown in [1], the temperature follows a an Auto-Regressive eXogenous (ARX) model where the inputs are the air conditioning state, fan speed, outside temperature, server load, free cooling damper state and temperature setpoint.

IV. An Evolutionary Scheduling Algorithm

Evolutionary algorithms (EA) are non-deterministic methods that emulate the evolution of species in nature to solve optimization, search, and learning problems [3].

Multi-Objective EA (MOEA) have been applied to solve hard optimization problems, obtaining successful results when solving real-life problems in many research areas [4]. Unlike many traditional methods for multiobjective optimization, MOEA are able to find a set with various trade-off solutions in a single execution, since they work with a population of tentative solutions. MOEA must be designed taking into account two goals: i) to approximate the Pareto front and ii) to maintain diversity instead of converging to a reduced section of the Pareto front. A Pareto-based evolutionary search leads to the first goal, while the second one is accomplished by using specific techniques also used in multimodal function optimization (sharing, crowding, etc.).

MOEA are often hybridized by combining two or more methods to solve the same problem, this takes advantage of the features of each method to improve the efficiency or accuracy of the new hybrid algorithm.

In this paper we use the *Non-dominated Sorting Genetic Algorithm, version II* (NSGA-II) [4] hybridized with a Local Search (LS) algorithm [5]. NSGA-II is a popular MOEA that have successfully been applied in many application areas. We propose to use a LS algorithm for improving the accuracy of the solutions computed by the NSGA-II's evolutionary process.

A. NSGA-II implementation

Solution encoding: Each solution represents the amount of cooling power and server power to be used at each time step, encoded as an integer vector of $2K$ elements. The first K elements represent the cooling power and the second K elements represent server power. The server power is encoded directly as Watts, whereas the cooling power is encoded as an integer value representing three states: (a) *1–100*: free cooling mode is applied, and the value represents the fan speed as a percentage of its maximum; (b) 101–200: the air conditioning unit is assumed to be operating, and (c) 201–300: neither air conditioning nor free cooling are in operation.

Optimization functions: The functions to optimize correspond to the ones defined in Equation 1. No modifications are required to encode them in NSGA-II.

Evolutionary operators: We use a three-point crossover operator (points p_1,p_2, and p_3); p_1 is determined randomly in (1,K), p_2 is K and p_3 is $K+p_1$. This way, we ensure that portions representing the same time interval for both cooling and server power move together from parents to offspring.

Mutation is applied to each gene with probability p_M. For a cooling power gene (position 1 to K), we replace its value v with $mod(v+rand()\times$MAX_HVAC,MAX_HVAC)). For the other genes, we redefine them with a random value between 0 and the maximum server power (i.e. all servers turned on).

Task scheduling algorithms: NSGA-II schedules the cooling power and the server power, but does not schedules the tasks execution in the datacenter. Given a cooling and server power, NSGA-II uses a subordinate algorithm for task scheduling. In this work we use the BFH greedy algorithm for task scheduling.

For the experimental analysis we configured NSGA-II with a population of 50 individuals, a crossover probability of 0.9, a mutation probability of 0.01, and a stopping criterion of 500 generations.

B. Greedy task-scheduling algorithm

We adapted the Best Fit Hole (BFH) greedy scheduling algorithm [1] to the specific features of the problem addressed in this article.

The BFH algorithm works as follows. First it sorts tasks according to their arrival times, and then it assigns them in order, filling the holes in the schedule of each computing resource. If a task fits into more than one hole, it is assigned to the hole that minimizes the difference between the hole duration and task's execution time. When no hole is available to execute a task, it is assigned to the machine that provides the minimum finishing time.

C. LS task-scheduling algorithm

We propose a new LS task-scheduling algorithm. LS algorithms are one of the most successful general approaches for finding high quality solutions for hard combinatorial optimization problems in reasonable time [5].

The proposed LS task-scheduler starts by generating an initial solution using the BFH algorithm. The neighbourhood for the LS is constructed using a simple task moving operation which moves one task from its current machine to a new machine in some arbitrary position. We use dominance as an acceptance criterion, that is, we accept a new best solution only when it dominates the current best solution. Finally, we use a stopping criterion of 4000 iterations.

We use a high-level relay hybridization schema [6]. First, the NSGA-II computes a set nondominated candidate solutions, and then we apply the LS to these solutions.

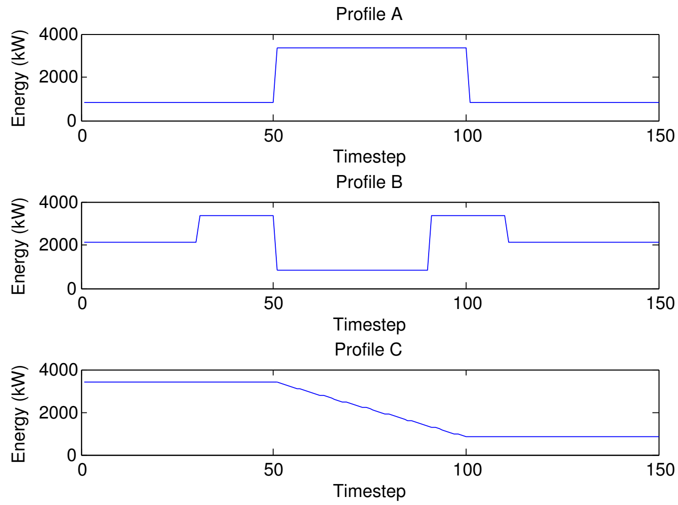

Fig. 2. Considered power profiles.

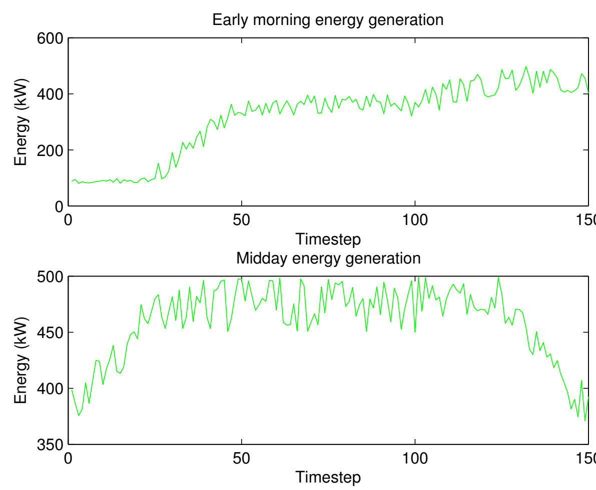

Fig. 3. Considered green power generation profiles.

V. Experimental Analysis

We use a set of problem instances to evaluate the proposed algorithms, each comprised of total of 150 1-minute timesteps. The task workload is comprised of 500 tasks, representing approximately 50% of the total computing capacity of the cluster. We consider three different power profiles, shown in Figure 2. Three different green energy generation profiles (A, B, and C) were also considered. Green profile A corresponds to an early morning power generation profile, green profile B corresponds to a midday power generation profile, and green profile C corresponds to a night time power generation profile (i.e. no green energy). Figure 3 shows green profile A and B. A total of 9 problem instances were evaluated combining the different energy profiles. The external temperature is fixed at 25°C. The initial temprature inside the datacenter is 26.5°C and the maximum allowed value is 27°C.

The datacenter in this work simulates the Parasol datacenter [2]. It consists of 64 Atom-based servers, each server

Green Profile	Respected Power Profile	Budget improvement	Deadlines met
A	89.47%	36.48%	42.61%
B	90.40%	44.28%	47.14%
C	70.77%	10.26%	36.96%

TABLE I
AVERAGE RESPECTED POWER PROFILE, BUDGET IMPROVEMENT, AND DEADLINES MET OF COMPROMISE SOLUTIONS COMPUTED BY NSGA-II.

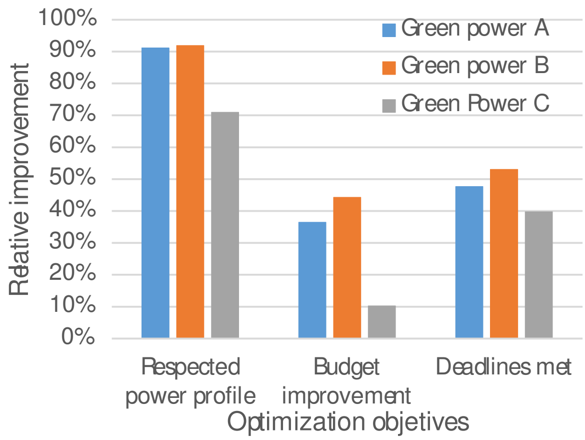

Fig. 4. Average improvements of the hybridized NSGA-II+LS.

consuming 30W in processing state, 22W in idle state, and 3W in sleep state. Considering all the servers and the HVAC system, the total maximum power consumption of Parasol is 4.2kW. For green power generation we simulated a solar panel array comprised of 3 solar panels capable of producing up to 0.5kW. Solar power generation for this work was generated using historical solar information from [2].

Table I presents average metrics of compromise solutions computed by NSGA-II. Preliminary results show that, in average, NSGA-II is able to respect the power profile objective **83.5%** of the time. When comparing with the Business As Usual (BAU) scenario, i.e. the same cluster with no green power sources and without scheduling machine state, the NSGA-II is able to reduce the budget by **30.3%** in average; reducing up to **44.3%** during daylight. Finally, in average, **42.2%** of the computing time of all tasks is executed within the deadline of its task.

Next, we evaluated the improvement of the proposed LS algorithm. Results show LS is able to further improve QoS by **11%** in average. This is a significant improvement, specially considering the low ratio of deadlines met by the compromise solutions computed by NSGA-II. On the other hand, power profile improvement is barely **1%** in average, and budget improvement is negligible. The low improvement in power profile and budget is expected, since most of the power is consumed/saved when scheduling the state of the machines. Figure 4 shows the aggregated average improvements of the hybridized NSGA-II+LS over the results computed for the BAU scenario. Figure 5 shows a sample Pareto front computed by NSGA-II and NSGA-II+LS.

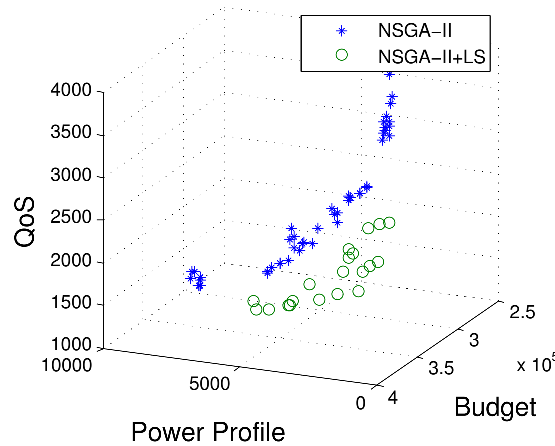

Fig. 5. Sample Pareto front computed by NSGA-II and NSGA-II+LS.

VI. Conclusion

This work introduces the problem of scheduling green-powered datacenters considering power consumption, budget and QoS, subject to maintaining the datacenter temprature below its operational threshold. We propose to schedule machines, tasks, and cooling devices simultaneously minimizing the deviation from a given power profile, the operational budget, and the amount of computing time violating the deadline for each task. We presented a mathematical formulation for the problem applying the datacenter model introduced in [1].

We proposed a NSGA-II hybridized with a LS for tackling the problem, and we evaluated it on a small number of problem instances representing different green power generation profiles and power profile thresholds. Preliminary results show the proposed method computes compromise solutions with a considerable budget reduction, low power profile deviation and a reasonable QoS level.

Future lines of work include, extending the scheduling time-horizon and improving the proposed method efficiency. Extending the scheduling time-horizon will enable us to consider tasks with far ahead deadlines. While improving the algorithm efficiency will enable us to reschedule more often, producing more accurate schedules.

References

[1] S. Nesmachnow, C. Perfumo, and I. Goiri, "Controlling datacenter power consumption while maintaining temperature and QoS levels," in 3^{rd} *IEEE International Conference on Cloud Networking*, 2014, pp. 242–247.

[2] I. Goiri, M. Haque, K. Le, R. Beauchea, T. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Matching renewable energy supply and demand in green datacenters," *Ad Hoc Networks*, vol. 25, Part B, no. 0, pp. 520–534, 2015.

[3] T. Bäck, D. Fogel, and Z. Michalewicz, Eds., *Handbook of evolutionary computation*. Oxford University Press, 1997.

[4] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. J. Wiley & Sons, Chichester, 2001.

[5] H. Hoos and T. Stützle, *Stochastic Local Search: Foundations & Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 2004.

[6] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, pp. 541–564, Sep. 2002.