# Infrastructure Aware Heterogeneous-Workloads Scheduling for Data Center Energy Cost Minimization

Kawsar Haghshenas, Somayyeh Taheri, Maziar Goudarzi, *senior Member*, *IEEE,* and Siamak Mohammadi, *senior Member*, *IEEE*

**Abstract**—Huge amount of energy consumption, the cost of this usage and environmental effects have become serious issues for commercial cloud providers. Solar energy is a promising clean energy source, to provide some portion of the Internet data center's (IDC's) energy usage which can reduce environmental effects and total energy costs. Moreover, due to the high energy consumption of the cooling system, considering cooling power in job scheduling can provide efficient solutions to reduce total energy consumption. In this paper, we investigate the problem of minimizing the energy cost of an IDC and propose an algorithm which schedules heterogeneous IDC workloads, by considering available renewable energy, cooling subsystem, and electricity rate structure. We evaluate the effectiveness and feasibility of our algorithm using real and synthetic workload traces. The simulation results illustrate how our proposed solution reduces data center energy cost by up to 46% compared to previous solutions. Moreover, results show that our solution is capable of reducing energy cost of data centers under different weather conditions, and rate structures.

**Index Terms**— Internet data center (IDC), Energy cost, Free cooling, Renewable energy, Uninterruptible power supply.

## 1 Introduction

THE scale of Internet Data Centers (IDCs) is increasing to meet the skyrocketing demand for IT applications and cloud services. In this regard, high energy consumption, the cost of this usage and environmental effects are critical issues that have received much attention in recent years. Estimated annual electricity costs of many IDC operators is more than $30 million [1] and in large IDCs, about 41% of the total operating costs is due to energy costs [2]. These costs and environmental effects of brown energy, motivate IDC owners to use renewable energy as much as possible. Some popular data centers such as Google [3] and Apple [4] have become pioneered in using renewable energy (Solar/Wind) up to 30% and 93% respectively, which is a huge improvement both economically and environmentally.

In a data center, resources must be cooled properly to ensure availability and reliability. This cooling process needs up to 30-50% of the total energy consumption of the data center [5, 6]. The most common cooling method in IDCs is mechanical air conditioning which requires considerable amounts of energy. Cooling energy consumption can be drastically decreased by using free cooling techniques. Airside free cooling technique uses outdoor air as a cold source for cooling and draws it inside directly [7]. The efficiency and applicability of airside free cooling are limited by moisture and impurities of the air. Furthermore, free cooling energy saving is highly sensitive to outdoor temperature variations so that, 2°C decrease in outdoor temperature can increase free cooling energy saving about 2-8% [8]. Thereby, in order to control the aforementioned dependencies to weather conditions, a parallel cooling scheme must be added to satisfy availability and reliability of the data center.

Workloads of the data centers are divided into two categories, service jobs, and batch jobs. The first one concerns about jobs that must be serviced in short time to satisfy predetermined Quality of Service (QoS). The second one is about the jobs with non-critical response time, where their process can be postponed according to the deadline, situation, and goal parameters of the scheduling algorithm. One opportunity to reduce data center energy cost is workloads scheduling, both temporally and spatially. There is an extensive research on workload scheduling in data centers. We know that, scheduling should guarantee Service Level Agreements (SLAs). Therefore, an important problem is scheduling and allocating service and batch jobs in a manner that resources are utilized efficiently, SLAs are met, and the total IDCs cost approaches to a reasonable minimum [9-15].

In common works on workload scheduling in green IDCs, jobs are scheduled based on the availability of renewable energy [9, 13, 16]. On the other hand, some thermal-aware scheduling approaches have been proposed based on minimizing the cooling energy consumption [10, 17].

Moreover, several studies, especially on geographical

- *K. Haghshenas is with the school of Electrical and Computer Engineering, University of Tehran, Tehran 17469-37181, Iran (email: khaghshenas@ut.ac.ir).*
- *S. Taheri and M. Goudarzi are with the Computer Engineering Department, Sharif University of Technology, Tehran 11155-9517, Iran (email: taheris@ce.sharif.edu, goudarzi@sharif.ir).*
- *S. Mohammadi is with the school of Electrical and Computer Engineering , University of Tehran, Tehran 17469-37181, Iran (email: smohamadi@ut.ac.ir).*

*Manuscript received November 4, 2017.*

distributed IDCs, explore the variety of rate structures to decrease the total energy cost of the data centers [18, 19].

Due to the variable nature of renewable energies, it is expected that scheduling algorithm postpones batch jobs to times with higher available renewable energy. However, this expectation may be changed by considering cooling power and energy storage system.

Based on this fact that, all above mentioned parameters affect the total energy cost of the data center, in this paper we propose an approach to overcome the drawbacks of the existing approaches in scheduling heterogonous workloads of the data center. The proposed approach is a three-step algorithm that schedules heterogeneous workloads in a green data center, powered by solar energy and grid, cooled by free cooling and mechanical chiller system. In the first step of this algorithm, available solar energy and incoming workloads are predicted. In the second step, the service job scheduling module allocates service jobs to the resources in a way that SLA can be guaranteed, and available resources for a predefined period of time are specified. Indeed, service jobs are acting as a baseline changing along the time. Considering service jobs helps to model a real data center, where available resources to run the batch jobs change along the time, which leads to an accurate evaluation of the algorithms. In the third step, the used amount of available solar energy, and batch job scheduling will be determined in order to minimize the total energy cost, considering a multi-electricity-market environment.

In most IDCs where the availability is important, there is Uninterruptible Power Supply (UPS) system with batteries to be used in the grid failure situation. These systems can be placed in a centralized or distributed manner, which there are real cases for each of these strategies [20, 21]. This energy storage is a power backup and is unused most of the time, thus it can be used beyond its original role. In [22] and [23], UPS has been applied to decrease the peak power consumption, and to save renewable energy, respectively. We investigate using distributed UPS system alongside with our proposed algorithm, in order to decrease energy cost, in two different scenarios described in Section 3.2.4.

In some previous works, the scheduling problem has been treated as an optimization problem and solved by known solutions like simulated annealing and other convex optimization solutions [9, 24]. Using optimization methods leads to more appropriate solutions; however, the main weakness of these approaches is their long run times especially for complicated problems.

The main contributions of this paper are as follows:

1) We propose a low-complexity algorithm to manage both service and batch jobs by considering the existence of renewable energy and free cooling, simultaneously. Furthermore, we investigate the effect of the weather condition on the efficiency of the proposed approach.

2) As many current IDCs are powered by both renewable and electrical grid, we consider both resources in workloads scheduling in order to minimize the total energy cost. Moreover, the proposed algorithm is feasible for any electrical rate structure and energy prices.

3) As emergency situations and consequently UPS batteries discharging, happen rarely, we investigate the usage of a UPS system beyond its main purpose, to decrease the total energy cost of an IDC.

4) The effectiveness of the proposed algorithm is evaluated using an in-house simulator tool written in C++. Simulation results demonstrate that our proposed algorithm reduces the total energy cost and energy consumption by up to 46% and 9%, respectively. The results of simulations for different weather conditions and rate structures show that the efficiency of our solution is independent of the used parameters (see Section 3). Furthermore, our results for synthetic benchmarks illustrate how our proposed approach distributes batch jobs along all 24-hour period, as much as possible. Finally, the results presented in Sec-

TABLE 1
NOTATIONS

| Notation | Description |
|---|---|
| $C(t)$ | Total energy cost at time slot t |
| $P(t)$ | Electricity price at time slot t |
| $IT(t)$ | IT computing equipment power consumption at time slot t |
| $IT_{max}$ | Maximum allowed IT power to preserve IT resources from damage |
| $CL(t)$ | Cooling equipment power consumption at time slot t |
| $CL_{air\text{-}side}(t)$ | Air side economizer power at time slot t |
| $CL_{chiller}(t)$ | Mechanical chiller power at time slot t |
| $\alpha$ | Constant coefficient of the air side economizer |
| COP | Coefficient of performance of the chiller |
| $T_S$ | Desired temperature after cooling |
| $Coef_{peak}(t)$ | Coefficient of peak power obtained from the grid at time slot t |
| $S(t)$ | Total charging power for the batteries at time slot t |
| $SU(t)$ | Total discharging power for the batteries at time slot t |
| $b^{ch}(t)$ | Whether the battery at time slot t is charging or not |
| $b^{d}(t)$ | Whether the battery at time slot t is discharging or not |
| $PV_{max}$ | Maximum solar power capacity |
| $PV(t)$ | Predicted solar power for time slot t |
| ${}_{used}PV(t)$ | Used solar power for IT or cooling resources at time slot t |
| $PV_{max}(t)$ | Predicted solar energy under sunny conditions for time slot t |
| $PV_{sold}(t)$ | Sold solar power at time slot t |
| $PV_{price}(t)$ | Solar energy price at time slot t |
| $CC(t)$ | Percentage of cloud cover predicted for time slot t |
| $G(t)$ | Total used grid power at time slot t |
| $\{SN\}$ | All of the incoming service jobs at the predefined time period $t_1$-$t_2$ |
| $\{BN\}$ | All of the incoming batch jobs at the predefined time period $t_1$-$t_2$ |
| SN | Number of All of the incoming service jobs at the predefined time period $t_1$-$t_2$ |
| BN | Number of All of the incoming batch jobs at the predefined time period $t_1$-$t_2$ |
| M | Number of time slots of predefined time period $t_1$-$t_2$ |
| $\{H\}$ | The set of all the hosts of the data center |
| K | Number of hosts |

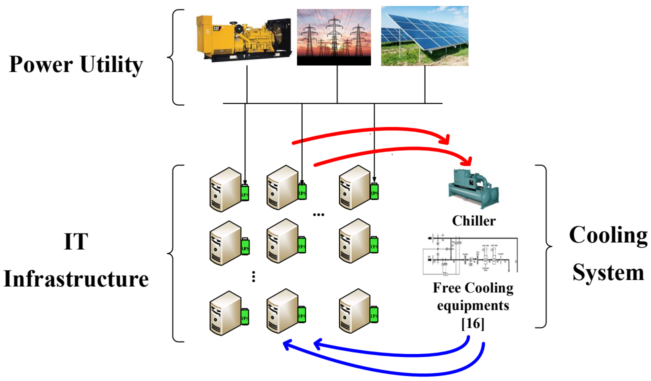


Fig. 1. A typical data center power, cooling and IT infrastructure

tion 3.3.4 show that utilizing UPS system along with the proposed approach, decreases energy cost by up to 4.4%.

The rest of this paper is organized as follows. Basic concepts, data center structure and problem formulation are introduced in Section 2. This section also represents our proposed algorithm. Section 3 evaluates the proposed algorithm with various case studies. The conclusion and future works are provided in Section 4.

## 2 System Model and Formulation

### 2.1 Data Center Structure

The data center contains a substantial amount of resources including a number of servers, batteries, cooling subsystem, network infrastructure, and power utility system. As mentioned before, UPS system in IDCs can be in the form of distributed, centralized or an intermediate level. In this paper, we consider a distributed UPS system depicted in Fig. 1. However, our solution is applicable to any UPS system placement approach. The applied cooling system includes free cooling and a chiller based system. Clearly, free cooling approaches are not actually free because they need a non-negligible amount of energy and an initial infrastructure. The assumed cooling approach is discussed in detail in the next section.

As the network is not in the scope of this paper, we have not shown it in the supposed data center structure. Data center power is provided by solar and grid energy. Morover, a diesel generator (DG) is employed as a backup power used in emergency situations. The whole system structure is illustrated in Fig. 1. For the sake of clarity, Table 1 lists the key notations used throughout this paper.

### 2.2 Workload Model

Let {SN} and {BN} represent the set of service and batch jobs that have to be scheduled at time slot t, respectively. These jobs include the jobs that arrive to the data center at the time slot t and the previously submitted jobs, which have not been scheduled yet. To keep it simple, we assume that all jobs arrive to the data center at the beginning of the time slots. The $j^{th}$ job which arrives to the data center at time slot t can be characterized by its resources requirement as well as its specifications. For this purpose, we introduce the tuple (j, $T_j$, t, $CPU_j$, $Mem_j$, $R_j$), where $CPU_j$ and $Mem_j$ represent computation resources and memory demand of the job, respectively. Also, $R_j$ denotes the total number of time slots that are needed to finish the job (in the case of allocating all requested resources). The binary parameter $T_j$ stands for the type of the job which can be zero for service jobs or 1 for batch jobs. In this paper, for the service jobs, we consider any delay in assigning the job to the servers as SLA violation. However, the batch jobs must be finished within a 24-hour time period after the submission time (see Section 2.4 for more details about the deadlines).

### 2.3 Cooling System

Current rapid increase in computing in data centers has resulted in an increase in the data center power density. Thereupon, an important challenge in optimizing IDCs is to minimize the requirements of the cooling system and its energy consumption.

A typical cooling system in a data center is the air conditioning system (Heating, Ventilation and Air Conditioning: HVAC). HVAC's main infrastructure that consumes power, consists of a chiller unit to chill the coolant material (water or air) and a circulation structure to direct cool air into the servers and remove heat from them [25]. The baseline case contains the computer room air conditioning (CRAC) units that are placed on the room floor. Hot air enters the top of the CRAC units and the cold air enters the room from the floor.

An effective technique to reduce cooling power is free cooling. In free cooling techniques, outdoor cool air is used instead of air which is cooled by chillers. Free cooling techniques include two categories, water-side free cooling, and air-side free cooling. In water-side free cooling, water is cooled by outdoor air and in air-side free cooling, outside air with a lower temperature than indoor is used to remove heat from servers directly.

Although free cooling is a highly effective way to improve IDC energy efficiency, its required weather condition leads to an integrated approach, where a reliable mechanical cooling system is available parallel to the free cooling system [26]. In this paper, we integrate air-side economizer, as a free cooling system, with the mechanical chiller. Chiller will be used if air side economizer is not sufficient to cool the data center. The air-side economizer scenario needs Air-Handling Units (AHU) to provide outdoor air for cooling.

Air-side economizer power consumption is the power consumed by the blower to direct the cool air into the server room. According to the fan basic laws [27], the blower power consumption is a cubic function of the blower speed [26]. On the other hand, the blower speed is proportional to the outside and inside temperature difference, where the inside temperature is a function of IT power. Hence, the air-side economizer power can be estimated by a cubic function of the IT power, as [24, 26]:

$$CL_{air-side}(t) = \alpha IT(t)^3 \tag{1}$$

where α is a coefficient that is proportional to the efficiency constant (Efficiency) and follows an inverse trend to the outside and inside temperature difference (ΔT):

$$\alpha = \text{Efficiency} / \Delta T^3 \tag{2}$$

The mechanical chiller power consumption is a function of IT power and is given by [28, 29]:

$$CL_{chiller}(t) = IT(t) / COP \tag{3}$$

The coefficient of performance (COP) in Equation (3) is the ratio of the amount of the removed heat by the CRAC unit to the total energy consumed by CRAC [30]. COP of a CRAC unit is a function of the desired temperature ($T_S$). In this paper, we use a typical COP model of the CRAC unit, utilized in the HP utility data center [28]. COP equation for this CRAC unit is calculated by:

$$COP(Ts) = 0.00068T_s^2 + 0.0008T_s + 0.458 \tag{4}$$

According to the considered hybrid cooling approach [24] in this paper, the optimal management of IDC cooling system is as follows: the data center is cooled by the air-side economizer as long as the outdoor and indoor temperature difference is high enough to cool down the data center with outside air (which corresponds to the IT power (IT (t)) consumption less than a predefined threshold) and then for a higher amount of IT (t), it needs to turn on the mechanical chiller to preserve reliably. Therefore, the data center cooling power is calculated as:

$$CL(t) = \begin{cases} \alpha IT(t)^3 & \text{if } IT(t) < E \\ \alpha E^3 + (IT(t) - E) / COP & \text{otherwise} \end{cases} \tag{5}$$

where E is the optimal cooling power capacity between the air-side economizer and the chiller. This value can be determined by [25]:

$$E = \sqrt{(\text{Efficiency} \times \Delta T^3) / COP} \tag{6}$$

It should be noticed that the α parameter in Equation (5) is a function of the outside and inside temperature difference (ΔT) and is calculated by Equation (2). Therefore, air side economizer power is also a function of (ΔT) through the $\alpha$ parameter.

### 2.4 Problem Definition and Cost Function

We propose a heuristic algorithm to manage data center resources, in order to minimize the total energy cost. We consider a data center equipped with distributed UPS and hybrid cooling system, and powered by solar and grid power. Therefore, we want to schedule the predicted incoming workloads including service and batch jobs. Also, we must determine battery and power utility operations for a predefined period of time.

Electricity rate structure is the method of calculating the cost of electricity. The rate structure is different in various regions and can affect the cost saving of using renewable resources. Mostly common rate structure types are flat, seasonal, time of use (TOU), and demand charges [31]. Rate structure often includes a combination of rate types. In this paper, we use the combination of TOU and peak demand charges. This is a billing scheme where the total energy cost is based on both the quantity and time of power usage. Each utility has its own method of billing its customers based on its policies and we could also include other charging policies. The impact of charging policies on the efficiency of the proposed algorithm will be investigated later in this paper. Let C (t) be the cost of energy at time slot t. Thereby, according to the considered rate structure, we have:

$$\begin{aligned} C(t) = {} & P(t) \times (IT(t) + CL(t) - PV(t)) + \\ & Coef_{peak}(t) \times \max_t (IT(t) + CL(t) - PV(t)) \end{aligned} \tag{7}$$

where P (t) and $Coef_{peak}$ (t) are the electricity price and the coefficient of peak power, at time slot t, respectively. IT (t), CL (t), and PV (t) are IT computing equipment power consumption at time slot t, cooling equipment power consumption at time slot t, and predicted solar power for time slot t, respectively. $Max_t$ (IT (t) +CL (t)-PV (t)) denotes the maximum consumed power from the grid, up to time slot t for a predefined time period. Thus, we formulate the cost minimization problem as:

$$\begin{aligned} \min \sum_{t=t1}^{t2} & P(t) \times (IT(t) + CL(t) - PV(t)) + \\ & Coef_{peak}(t) \times \max_t (IT(t) + CL(t) - PV(t)) \end{aligned} \tag{8a}$$

$$\text{s.t. } 0 < IT(t) < IT_{max} \tag{8b}$$

$$0 < PV(t) < PV_{max} \tag{8c}$$

$$\sum_{j=1}^{JN} CPU_i^j \le CPU_i \quad \forall i \in H \tag{8d}$$

$$\sum_{j=1}^{JN} Mem_i^j \le Mem_i \quad \forall i \in H \tag{8e}$$

$$\text{For all jobs in } \{SN\} \text{ and } \{BN\} \text{ SLA must be met.} \tag{8f}$$

where the constraints (8b) and (8c) ensure that the maximum IT power and maximum available solar power, for time slot t, have been considered. The third and fourth constraints guarantee that for all the hosts in {H}, the total allocated resources to its running jobs is lower than its maximum computational and memory capacities. $CPU_i^j$ and $Mem_i^j$ represent allocated CPU and memory resources of the $i^{th}$ host to the $j^{th}$ job.

Finally, the constraint (8f) guarantees that for all requests, SLA requirements are met. Clearly, for different request categories in our problem, the SLA requirements are different. For service jobs, the average delay (or 95th percentile delay) is usually used as the SLA metric, while a completion deadline is used for the SLA metric of batch jobs [23]. In this paper, for the service jobs, we consider any delay in assigning the jobs to the servers as SLA violation and the batch jobs must be finished within a 24-hour time period after the submission time. As our main goal in this paper is to minimize the energy cost, we consider QoS requirements as a constraint which has to be met. Therefore, in the experimental evaluation (see Section 3), for all considered algorithms the simulated data center is big enough to meet QoS requirements, except for "SLA violation discussion" at the end of Section 3.3.1, where we have simulated smaller data centers to evaluate SLA violation caused by the introduced algorithms.

In recent years, the environmental effects of data centers have been increased and highly regarded. On the other hand, the solar panel price is significantly reducing [32]. Therefore, it can be imagined that in the near future, most of the IDCs will be powered by solar energy powerhouse that, in addition to supply IDC power requirements, will be able to have extra solar energy in some time slots. Also, in some countries, in order to increase the usage of renewable energies, the renewable energy selling price to the grid, is higher than the grid energy price. These rates are based on the utility policies. More specifically, local flexibility markets as a market-based management mechanism for aggregators has provided the opportunity of self-balancing of locally generated energy for providers [33]. Therefore, the ability of selling and buying some portion of solar power is added to the basic formulation of the energy cost calculation.

Let $PV_{used}$ (t) represents the used solar power at time slot t while $PV_{sold}$ (t) represents the sold solar power to the grid at time slot t. Thus, if we add the opportunity of selling extra solar energy to the described problem, Equation (8) can be written as:

$$\min \sum_{t=t_1}^{t_2} P(t)\times(IT(t)+CL(t)-PV_{used}(t)) + Coef_{peak}(t)\times \max_t(IT(t)+CL(t)-PV_{used}(t)) - PV_{sold}(t)\times PV_{price}(t) \quad (9a)$$

$$\text{s.t.} \quad 0 < IT(t) < IT_{max} \quad (9b)$$

$$0 < PV(t) < PV_{max} \quad (9c)$$

$$\sum_{j=1}^{JN} CPU_i^j \le CPU_i \quad \forall i \in H \quad (9d)$$

$$\sum_{j=1}^{JN} Mem_i^j \le Mem_i \quad \forall i \in H \quad (9e)$$

$$0 < PV_{used}(t) < IT(t)+CL(t) \quad (9f)$$

$$PV_{used}(t) + PV_{sold}(t) = PV(t) \quad (9g)$$

$$\text{For all jobs in } \{SN\} \text{ and } \{BN\} \text{ SLA must be met.} \quad (9h)$$

While optimal job scheduling (considering different parameters) is an NP-complete problem, we propose a three step heuristic algorithm with low computational overhead (its time complexity is a linear function of the number of jobs (see Section 2.5)) that can be applied in real-time for described problem.

## 2.5 Proposed Algorithm

In this section, we propose a three-step algorithm (Algorithm 1) for the problem described in the previous section. In this algorithm, we schedule jobs based on the predicted solar energy and weather conditions for the next time period. We need to predict the available solar energy and the incoming workloads to the data center in the future. Therefore, in the first step of the proposed algorithm, we predict the available solar power for the next time slots ($t_1$ – $t_2$ time period) by the model presented in [34]. This model which has been used widely in the previous works is as follows:

$$PV(t) = PV_{max}(t)\ (1-CC(t)) \quad (10)$$

where, $PV_{max}$ (t) is the maximum possible solar power from a specific solar panel size, in a given region, and CC(t) is the predicted cloud cover from meteorological information. There are many solutions to predict the incoming workloads to data centers. As workload prediction is not in the scope of this paper, we use some real workload traces available in [35] and [36].

In the next step of Algorithm 1, according to the utilization of resources and a special scheduling algorithm, service jobs are scheduled and allocated to the hosts, so that their SLA are met and free capacity of hosts is specified for the next time slots.

There are many algorithms to schedule service jobs. In this paper, we schedule service jobs based on the FIFO approach. Moreover, we use best-fit algorithm to allocate virtual machines to the hosts. The best-fit algorithm allocates a virtual machine to the host with the closest free resources (CPU and memory). This means that minimal empty space will be left in the target host.

At this point of the proposed algorithm, the utilization of resources, the available solar energy, and the incoming batch jobs are given for the next time slots of the predefined time period.

In the third step of Algorithm 1, we schedule batch jobs in order to minimize the total energy cost. As mentioned before, selling extra solar energy can be efficient in some time slots. Thus in this step, we also determine how to use the predicted solar energy. Indeed, the third step of the proposed algorithm consists of two sub-steps. At the first sub-step (3.A in Algorithm 1), for any time slot of the future time period, the amount of selling and using solar energy is determined. Total energy cost at time slot t is calculated by Equation (7). If we have the opportunity of selling solar energy, the energy cost of time period t is calculated by:

$$C(t) = P(t)\times(IT(t)+CL(t)-PV_{used}(t)) + Coef_{peak}(t)\times \max_t(IT(t)+CL(t)-PV_{used}(t)) - PV_{sold}(t)\times PV_{price}(t) \quad (11)$$

In order to determine $PV_{used}$ (t) and $PV_{sold}$ (t), we use the following equation:

$$P(t)\times(IT(t)+CL(t)-PV_{used}(t)) + Coef_{peak}(t)\times \max_t(IT(t)+CL(t)-PV_{used}(t)) - PV_{sold}(t)\times PV_{price}(t) \le P(t)\times(IT(t)+CL(t)-PV(t)) + Coef_{peak}(t)\times \max_t(IT(t)+CL(t)-PV(t)) \quad (12)$$

The first part of this equation calculates the energy cost in the case of selling a fraction of the available solar energy, which is compared to the energy cost in the case of using total available solar energy. This equation is initiated with $PV_{used}$ (t) = PV (t) and $PV_{sold}$ (t) = 0. After initialization, $PV_{used}$ (t) is decreased from PV (t) and $PV_{sold}$ (t) is increased from zero gradually until inequality is violated or $PV_{sold}(t) = PV(t)$.

In the second sub-step (3.B in Algorithm 1), the biggest unscheduled batch job will be scheduled to minimize total energy cost. The determined time slot for running each batch job affects the total energy cost.

As the calculation of sold and used solar energy in the first sub-step of step three, is based on the energy cost, the result of the second sub-step of step three can affect the optimum decision of the first sub-step. Therefore, the first substep is repeated, after scheduling each batch job.

**Algorithm 1**, batch job scheduling algorithm

1. Predict the available solar energy for next time slots (from $t_1$-$t_2$), ***PV (t)*** and incoming workloads, ***{SN*** and ***{BN}***.
2. Schedule and allocate all jobs in ***{SN*** by FIFO approach and best-fit algorithm respectively.
3. Schedule all jobs in ***{BN}*** and determine using solar energy for next time slots.

   A. Calculate and schedule solar power for next time slots, ***t***.

   **if** ($PV_{price}(t) \le P(t)$)
   $PV_{used}(t) = PV(t)$;
   $PV_{sold}(t) = 0$;
   **end**
   **else**
   Use Equation (12) to decide $PV_{used}(t)$ and $PV_{sold}(t)$.
   **end**

   B. **while** (all batch jobs are not scheduled)
   ***currentBatch*** = the biggest unscheduled batch job;
   ***CBRunTime***, determined time slot to run ***currentBatch;***
   **for** each $timeSlot_i$ of the predefined time period,
   **if** ***currentBatch***'s deadline is met
   ***CBRunTime*** = $timeSlot_i$ ;
   Calculate total energy cost;
   **end**
   **end**
   ***CBRunTime*** = choose $timeSlot_i$ with minimum total energy cost.
   **Do** sub-step 3.A.
   **end while**

Let M represents the number of next time slots considered in the proposed algorithm, and K the number of hosts in the data center. BN and SN are the numbers of batch jobs and service jobs of the next M time slots. The complexity analysis of the proposed algorithm and its basic operations are as follows:

1. Predicting the available solar energy for next time slots: $O(M)$
2. Scheduling all service jobs by FIFO: $O(1)$
3. Allocating all service jobs by best-fit: $O(SN \times K)$
4. Scheduling all jobs in {BN} and determining used solar energy for next time slots: $O(BN \times 2M)$
   - Calculating and scheduling solar energy for all next time slots: $O(M)$
   - Scheduling biggest unscheduled batch job: $O(M)$

Total complexity of the proposed algorithm is:

$$M + 1 + (SN \times K) + (BN \times 2M) \tag{13}$$

The number of servers and considered next time slots are constant values. Therfore, overall complexity of Algorithm 1 is:

$$O(SN + BN) \tag{14}$$

# 3 PERFORMANCE EVALUATION

In this section, we evaluate our proposed algorithm by simulation results. We compare our proposed algorithm with some existing algorithms, and investigate the effect of various parameters on the efficiency of the algorithm.

## 3.1 Experimental Setup

To evaluate the effectiveness of our solution, we use an in-house simulator tool written in C++, where we have coded the proposed and compared algorithms on top of it. A computer with a Core i7, 3.6 GHz CPU and 32GB RAM has been used to run the simulations. We consider a data center with 100 and a data center with 400 Physical Machines (PMs). The former is used for benchmarks with at most 500 jobs and the next one is used for bigger benchmarks. The server model used in simulations is DL360 G7 [37] with 12 core 2.67GHz Intel processors and 12GB RAM. The idle and maximum power consumption of this server are 103 and 270.9 W. The CPU utilization of a server is a good estimation for its power consumption [38]. Therefore, we utilized the most used power model [17, 39] which is:

$$P_{server} = \begin{cases} P_{idle} + (P_{max} - P_{idle}) \times U & \text{if } U>0 \\ 0 & \text{if } U=0 \end{cases} \tag{15}$$

where $P_{idle}$ and $P_{max}$ are the power consumption of the server at the idle and full utilization modes respectively and U represents the server CPU utilization. The energy is the product of the power and its time duration.

Used workloads in this paper are 24-hour traces from NASA-iPSC, LANL-O2K [35], and grid5000_clean _ trace [36] as real benchmarks. In these benchmarks, which have been represented in Standard Workload Format (SWF), the jobs with QueueID = 0 are service jobs (which usually are the jobs with lower computational overhead) and the jobs with QueueID > 0 are the batch jobs (with higher computational overhead, higher run time and lower priority). Totally, we use 6 various workload traces to test the proposed algorithm. The characteristics of these benchmarks are shown in Table 2. In our experiments, we used Equation (10) to predict future available solar power. $PV_{max}$ (t) in this equation, is dependent on the region and capacity of the solar power station. In this paper, for each benchmark, we consider a solar power station that can provide about 30% of total data center's energy consumption. The parameter CC(t) is provided by several meteorology tools such as the tool available at [40]. After providing needed data, we use PV Watts Calculator available at [41] to predict future available solar energy.

## 3.2 Algorithms in Comparison

We consider four algorithms to compare with the

TABLE 2
REAL WORKLOAD TRACES FOR EXPERIMENTS

| Benchmark | Number of jobs (#) | Batch jobs percentage (%) | Source benchmark |
|---|---|---|---|
| **Benchmark #1** | 129 | 41.8 | grid5000_clean_trace [36] |
| **Benchmark #2** | 342 | 73.9 | NASA-iPSC [35] |
| **Benchmark #3** | 379 | 37.7 | NASA-iPSC [35] |
| **Benchmark #4** | 517 | 47 | LANL-O2K [35] |
| **Benchmark #5** | 742 | 65.9 | LANL-O2K [35] |
| **Benchmark #6** | 882 | 55.4 | LANL-O2K [35] |

TABLE 3
ALGORITHMS EXECUTION TIME (IN SECONDS)

| Benchmark \ Algorithm | BSBGA | Proposed | ASAP | Day | Night |
|---|---|---|---|---|---|
| **Benchmark #1** | 406.1 | 9.79 | 0.16 | 0.82 | 0.44 |
| **Benchmark #2** | 798.92 | 59.85 | 0.3 | 9.13 | 7.35 |
| **Benchmark #3** | 546.54 | 41.41 | 0.34 | 5.17 | 3.3 |
| **Benchmark #4** | 3600.9 | 60.4 | 0.28 | 3.71 | 9.9 |
| **Benchmark #5** | 3613.9 | 189 | 0.42 | 10 | 23 |
| **Benchmark #6** | 4209.6 | 560 | 0.45 | 14 | 46 |

proposed algorithm, ASAP, Night, Day and BSBGA:

1) ASAP: the simplest and least-used algorithm is as soon as possible (ASAP) algorithm. ASAP schedules batch jobs just like service jobs and does not use the opportunity of non-criticality of batch jobs to decrease energy cost. This approach can increase service jobs SLA violation.
2) Night: one of the most-used algorithms is night algorithm. This algorithm benefits from lower electricity price (in most rate structures) and lower temperature at night. Moreover, according to minimum incoming service jobs at night hours, night algorithm leads to the minimum conflict between service and batch jobs.
3) Day: another rudimentary algorithm is day algorithm. This algorithm runs batch jobs at times with maximum available solar energy. Therefore, Day algorithm is a common algorithm in data centers powered by solar energy. This algorithm also may turn off more servers at night hours.
4) BSBGA: the problem of scheduling batch jobs, formulated in section 2, is an optimization problem. Therefore, known convex optimization algorithms such as genetic algorithm [42], can be used to solve it. In this paper, we consider batch job scheduling based on genetic algorithm (BSBGA) as one of the comparing algorithms. We have implemented this algorithm using Matlab optimization tools. In this way, the goal parameter is the same as Equation (8a) (total energy cost), and the applied constraints are (8b), (8c), (8d), (8e), and (8f). We use the results of the genetic algorithm as the best answer to compare with our proposed algorithm.

### 3.3 Simulation Results

#### 3.3.1 Energy Consumption, Energy Cost, and SLA Violation Evaluation

We begin an evaluation by calculating energy consumption and total energy cost of the data center, for proposed and compared algorithms. Fig. 2 demonstrates the total energy cost for introduced algorithms running various benchmarks. As shown in this figure, compared to other algorithms except for BSBGA, the proposed algorithm leads to minimum lower energy cost. Furthermore, for the larger benchmark, the improvement percentage will be further improved. Proposed algorithm improves energy cost 2-9% for Benchmark #1 and 15-29% for Benchmark #6 compared to other approaches. Minimum energy

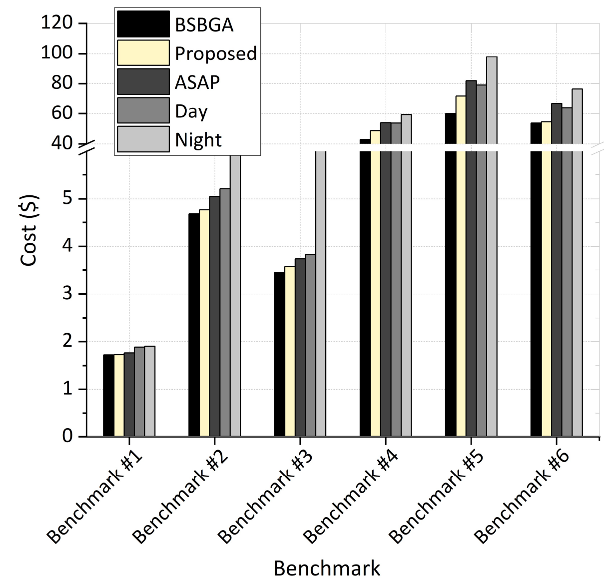


Fig. 2. Total energy cost for introduced algorithms

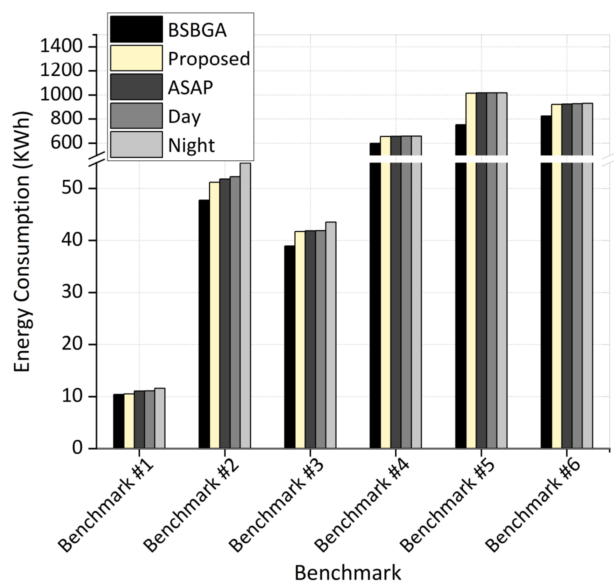


Fig. 3. Total energy consumption for introduced algorithms

cost is from BSBGA algorithm but the main flaw of this algorithm is its huge runtime especially for bigger benchmarks. BSBGA algorithm reduces energy cost 0.1-19% compared to the proposed algorithm. This improvement is not proportional to the number of jobs. Therefore, using BSBGA for bigger benchmarks, according to its much longer runtime, is even worse.

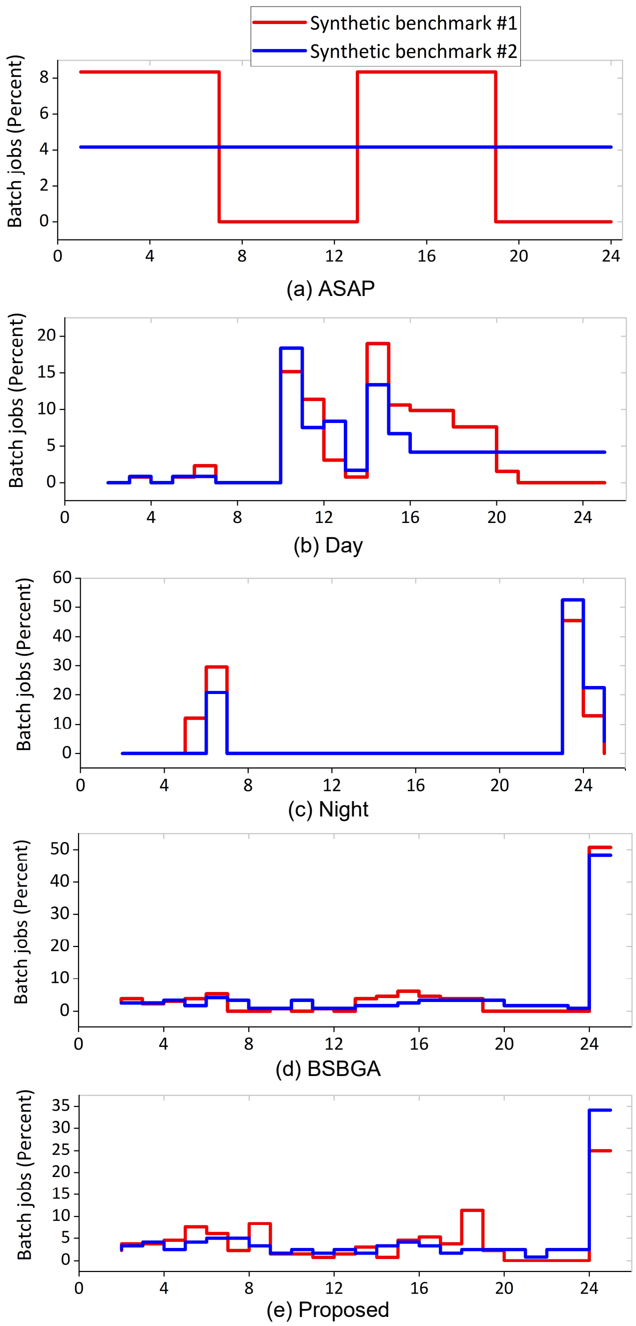


Fig. 4. The execution time of batch jobs of Synthetic benchmarks, scheduled by (a) ASAP algorithm. (b) Day algorithm. (c) Night algorithm. (d) BSBGA algorithm. (e) Proposed algorithm.

Table 3 represents the runtime of the introduced algorithms. It is seen that the proposed algorithm takes much less time than BSBGA because of its smaller complexity. As we mentioned in the previous section, the proposed algorithm's runtime is proportional to the number of service and batch jobs (we can see this from Table 3, too).

As Fig. 2 shows, night algorithm leads to maximum energy cost. That is because night algorithm uses minimum solar energy and runs batch jobs during the night hours whose service jobs are minimum. As the 32% of the service jobs in Benchmark #4 come to the data center at night, night algorithm leads to a better server consolidation for this benchmark compared to other benchmarks.

TABLE 4
AVERAGE NUMBER OF SLA VIOLATIONS PER MINUTE

| Benchmark \ Algorithm | BSBGA | Proposed | ASAP | Day | Night |
|---|---|---|---|---|---|
| **Benchmark #1** | 0 | 0 | 0 | 0 | 0 |
| **Benchmark #2** | 0.14 | 0.12 | 0.14 | 0.13 | 0.1 |
| **Benchmark #3** | 0.27 | 0.29 | 0.29 | 0.29 | 0.26 |
| **Benchmark #4** | 11.29 | 11.14 | 11.29 | 11.28 | 11.04 |
| **Benchmark #5** | 25.01 | 24.27 | 25.73 | 24.9 | 23.06 |
| **Benchmark #6** | 3.04 | 3.06 | 3.11 | 3.11 | 2.975 |

The total energy consumption of introduced algorithms for various benchmarks is shown in Fig. 3. The proposed algorithm not only has decreased total energy cost compared to other algorithms, but also it has not increased the energy consumption and even has decreased in some cases. This is because the proposed algorithm considers effective parameters simultaneously. Totally, the efficiency of day, night and ASAP algorithms completely depends on the incoming workload pattern and environmental conditions but the proposed algorithm achieves the highest energy cost saving for all workloads.

Presented results are from real workload traces. In order to observe the operation of each algorithm, we use two other benchmarks that have been built from Benchmark #3. We name these benchmarks Synthetic Benchmark #1 and Synthetic Benchmark #2. Half of the batch jobs of Synthetic Benchmark #1 enter the data center during the first 6 hours of 24-hour and other ones enter during third 6 hours of 24-hour. In Synthetic Benchmark #2, all the batch jobs, are uniformly distributed along 24 hours. The percentage of batch jobs of synthetic benchmarks, scheduled using various algorithms, is shown in Fig. 4. More specifically, the vertical axis represents the percentage of the batch jobs scheduled at each hour of 24-hour time period. As this figure shows, our proposed approach distributes batch jobs along all 24-hour period, as much as possible.

## SLA Violation discussion:

In order to evaluate SLA violation of the data center due to the different algorithms, the simulations are repeated for the data centers with 10 and 50 PMs (for benchmarks with at most 500 jobs and for bigger benchmarks, respectively). In this regard, for the service jobs, after submission time, and for the batch jobs, after missing the deadline (every batch job must be finished within a 24-hour after submission), every 1-minute delay in assigning the job is considered as an SLA violation. Table 4 shows the average number of SLA violations per minute, caused by different algorithms over a 24-hour of the data center operation.

As we can see from Table 4, night algorithm causes the minimum number of SLA violations as it moves the batch jobs to the night hours, where the service jobs are minimum. It should be noticed that, in the implementation of the proposed and compared algorithms, the available resources are identified and the batch jobs are rescheduled based on them. In this way, the caused SLA violations of these algorithms cannot be higher than ASAP.

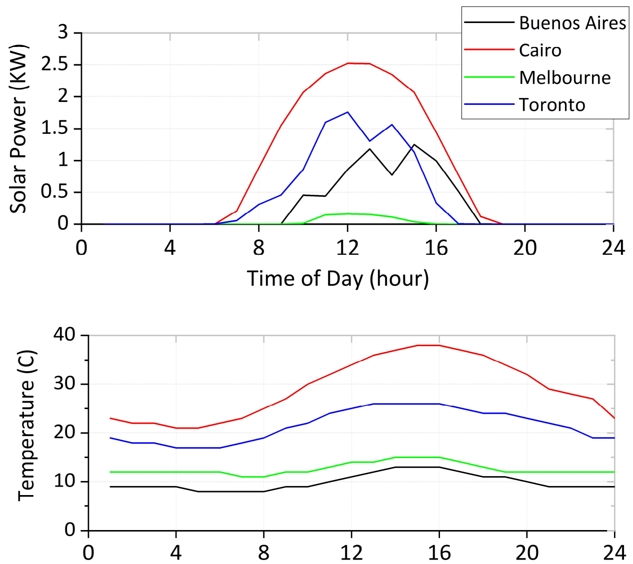


Fig. 5. A typical temperature and predicted available solar power for chosen regions

#### 3.3.2 Impact of Weather Condition

In this section, we evaluate our proposed algorithm, in various geographical conditions. Various weather conditions affect the predicted solar energy and free cooling capacity. For this investigation, we have chosen four cities; Buenos Aires, Toronto, Cairo, and Melbourne that are in different geographical locations which leads to different weather conditions. A typical predicted temperature [40] and predicted available solar energy [41] for these regions, are shown in Fig. 5.

Total energy cost under different algorithms and benchmarks, for a 24-hour time period, is illustrated in Fig. 6. As we can see from this figure, our proposed algorithm achieves the minimum energy cost compared to other algorithms (except BSBGA) in all cases.

The proposed algorithm reduces energy cost 2-22%, 2-24% and 6-46% compared to ASAP, Day, and Night algorithms, respectively. The maximum and minimum improvements of the proposed algorithm is for Cairo and Melbourne weather conditions, respectively. This outcome is related to the available solar energy and weather condition. Therefore, it may change by changing the capacity of the solar powerhouse.

#### 3.3.3 Impact of the rate structure

As mentioned before, electricity rate structure may affect the efficiency of scheduling approaches. The correlation of different rate structures with the available solar energy and free cooling potential is different. Therefore, in this section, we investigate the efficiency of the proposed algorithm in different rate structures. In the previous sections of this paper, we have used a combination of main rate types, TOU and demand charges, that is:

$$C(t) = P(t) \times G(t) + Coef_{peak}(t) \times \max G(t) \tag{16}$$

where maxG (t) is the maximum grid power used by the data center for a predefined time period. In this section, we consider two more rate structures whose characteristics are shown in Table 5.

The first rate structure is predefined rate structure in Equation (16). The second rate structure is a TOU with different off-peak, mid-peak and on-peak hours from the first rate structure.

TABLE 5
DESCRIPTION OF CONSIDERED RATE STRUCTURES

| Rate structure | Structure |
|---|---|
| TOU& Maximum Usage | TOU & demand charges - 1-8 and 24 are off-peak, 9-19 mid-peak and 20-23 on-peak [43] and Equation (16) |
| TOU | TOU- 1-8 and 24 are off-peak, 9-19 mid-peak and 20-23 on-peak [43] |
| TOU& Grid Power Limit | TOU & demand charges- 1-5 and 22-24 off-peak, 10-16 and 20-21 mid-peak and 6-9 and 17-19 on-peak [44] and Equation (17) |

The equation used for the third rate structure is:

$$C(t) = \begin{cases} P(t) \times G(t) & \text{if } G(t) < PEAK \\ P(t) \times G(t) + & \\ P'(t) \times (G(t) - PEAK) & \text{otherwise} \end{cases} \tag{17}$$

where PEAK and P′(t) are the predefined grid power limit and predefined price for extra grid power usage from PEAK. In this rate structure, if the power usage from the grid at time slot t, is less than the PEAK, the energy price is P (t) and for the power consumption higher than the PEAK, the energy price would be P′(t).

Total energy cost of the data center for a 24-hour time period, under introduced rate structures, is shown in Fig. 7. As we can see from this figure, the total energy cost of the data center can be varied widely under different electricity markets with different rate types. In addition, the proposed algorithm keeps its efficiency for all considered rate structures while other approaches may operate efficiently in only limited situations.

The results of sections 3.2.2 and 3.2.3 argue that the proposed algorithm can be applied to distributed multi data centers too. These distributed DCs can be located in various geographical regions with different electricity markets.

#### 3.3.4 Using UPS in order to Decrease Energy Cost

As mentioned before, nearly all IDCs contain UPS system as a backup power option. This UPS system creates an opportunity to reduce energy cost by providing energy storage system. By considering this opportunity, optimization problem in Equation (9), can be written as:

$$\min \sum_{t=t_1}^{t_2} P(t) \times (IT(t) + CL(t) + S(t) - PV_{used}(t) - SU(t)) + Coef_{peak}(t) \times \max_t (IT(t) + CL(t) + S(t) - PV_{used}(t) - SU(t)) - PV_{sold}(t) \times PV_{price}(t) \tag{18a}$$

$$\text{s.t. } 0 < IT(t) < IT_{max} \tag{18b}$$

$$0 < PV(t) < PV_{max} \tag{18c}$$

$$\sum_{j=1}^{JN} CPU_i^j \le CPU_i \quad \forall i \in H \tag{18d}$$

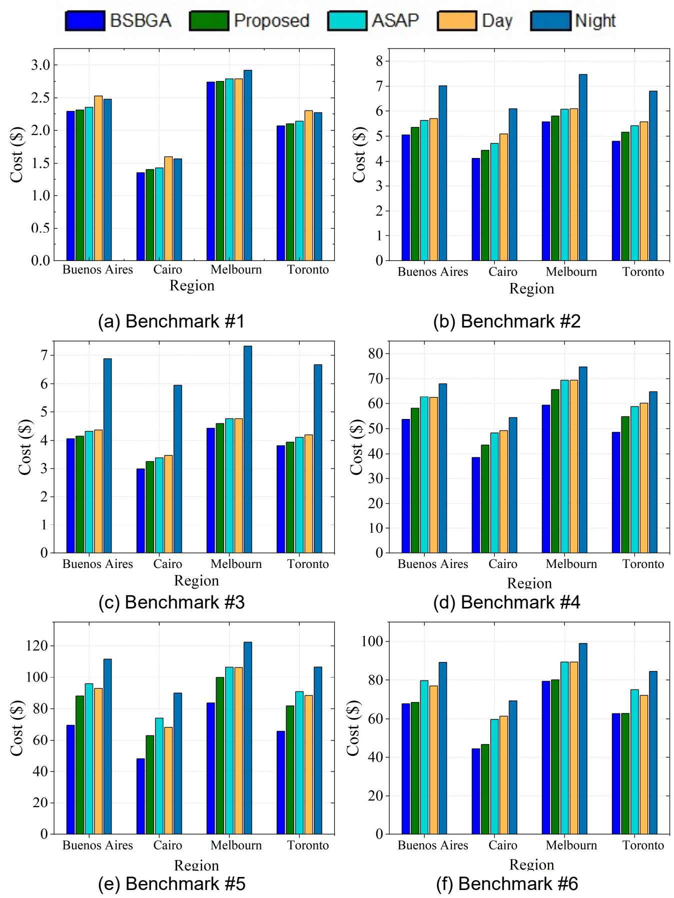


Fig. 6. Total energy cost for different regions for (a) Benchmark #1. (b) Benchmark #2. (c) Benchmark #3. (d) Benchmark #4. (e) Benchmark #5. (f) Benchmark #6.

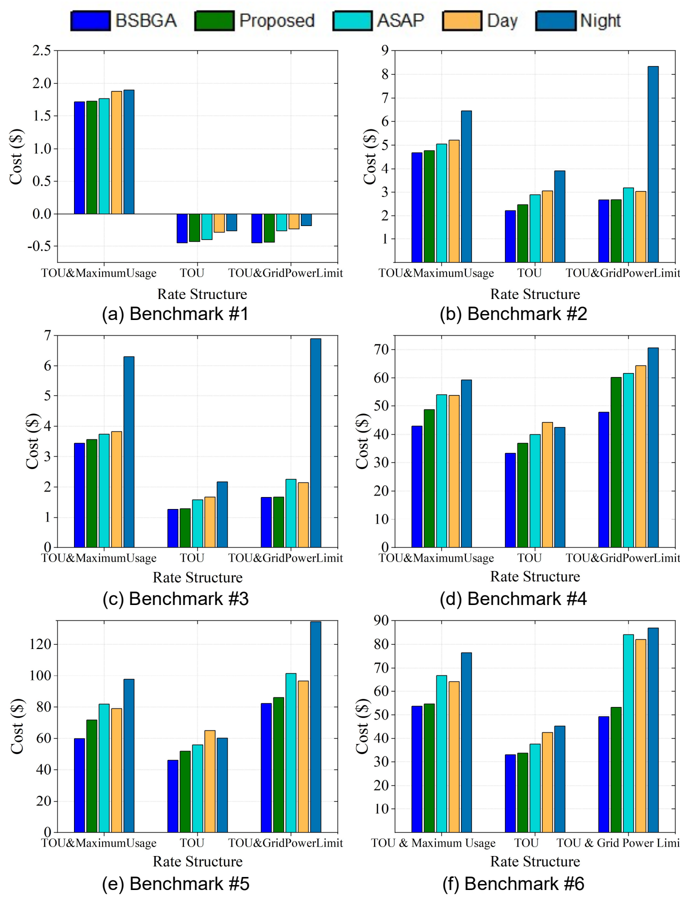


Fig. 7. Total energy cost for different rate structures for (a) Benchmark #1. (b) Benchmark #2. (c) Benchmark #3. (d) Benchmark #4. (e) Benchmark #5. (f) Benchmark #6.

$$\sum_{j=1}^{JN} Mem_i^j \leq Mem_i \qquad \forall i \in H \tag{18e}$$

$$0 < PV_{used}(t) < IT(t) + CL(t) \tag{18f}$$

$$PV_{used}(t) + PV_{sold}(t) = PV(t) \tag{18g}$$

$$\text{For all jobs in } \{SN\} \text{ and } \{BN\} \text{ SLA be met.} \tag{18h}$$

For all batteries,

$$b^{ch}(t), b^{d}(t) \in \{0,1\} \tag{18i}$$

$$b^{ch}(t) + b^{d}(t) \leq 1$$

where S (t) and SU (t) are the total charging and discharging power for all batteries at time slot t, respectively. In addition, for each battery, to indicate the battery is charging or discharging, two binary variables, $b^{ch}$ (t) and $b^{d}$ (t) are used where $b^{ch}(t)$ =1 if the battery is charging and $b^{d}$ (t) =1 if the battery is discharging.

We assume that for each battery, charging and discharging cannot be done simultaneously. Therefore, at each time slot t, the sum of $b^{ch}$ (t) and $b^{d}$ (t) has to be less or equal to one. In this way the total charging and discharging power for all batteries at time slot t is calculated as follows:

$$S(t) = \sum_{i=1}^{K} b_i^{ch}(t) \times charge_i(t) \tag{19}$$

$$SU(t) = \sum_{i=1}^{K} b_i^{d}(t) \times discharge_i(t) \tag{20}$$

where $charge_i(t)$ and $discharge_i(t)$ represent the amount of charging and discharging power for the $i^{th}$ battery at time slot t, respectively. It should be noticed that we do not consider some practical issues such as energy leakage in the battery or DC/AC conversion loss.

Algorithm 2, is a battery charging plan which consists of two main steps. Each battery with uncharged capacity, will be charged in two cases; there is more solar power than the DC power consumption and the price of grid power is more than solar power or time slot t is off-peak based on the electricity rate structure.

In the case of using UPS system to decrease energy cost, we consider two different scenarios for the accessibility of PMs to batteries. The first one assumes that any PM can use all batteries in the data center. Indeed, in the first scenario, distributed UPS acts like centralized UPS. In the second scenario, each PM only can use its own battery. Indeed, in this section, we investigate the impact of using batteries to decrease energy cost and compare the mentioned scenarios of this usage. In this regard, we use Algorithm 3 to allocate jobs to hosts based on their free resource capacities and their battery charges. Algorithm 3 is a modified best-fit algorithm that the definition of "best" is updated. This algorithm selects a host for a job if

TABLE 6
TOTAL ENERGY COST FOR DIFFERENT SCENARIOS

| Scenario | Total energy cost ($) |
|---|---|
| Proposed algorithm | 3.57 |
| Proposed algorithm / centralized UPS | 3.41 |
| Proposed algorithm / distributed UPS | 3.51 |

it fits job's required resources and its allocated battery has the maximum available electricity charge for the next time slot (steps 2 and 3 of the Algorithm 3).

**Algorithm 2**, Charging Plan for batteries

Input: ***Battery<sub></sub>*** 

```
Input: Battery_i
Output: Charging Plan
1. for each timeSlot_j of the predefined period,
     if IT(t) + CL(t) < PV(t)
       if P(t) > PV_price(t)
           charge the battery
           Break
       end
     end
2. for each timeSlot_j of the predefined period,
     if P(t) is off-Peak
       charge the battery
       Break
     end
3. Return charging Plan.
```

The batteries will be discharged based on the host requirement that is impressed by Algorithm 3. We have assumed that each battery can be charged and discharged at most once in the pre-defined time period from $t_1$-$t_2$. This limitation can be justified based on the rate structure, battery model, and available solar energy.

**Algorithm 3**, Job Allocation

```
Input: Job_i
Output: Selected Host
1. for each hostj in host list L,
     if Job_i fits hostj
     Calculate remaining capacity after allocation.
     end
   end
2. Select all hosts with minimum remaining capacity
   after allocation, host list L1.
3. Select a host in host list L1 with maximum battery
   charge, Selected Host.
4. Return Selected Host.
```

In this paper, we considered a 900W UPS beside each PM. This UPS can provide 6-minute maximum load and 12-minute half load. As the server consumes 207.9W at its maximum load, the 900W UPS is enough to guarantee availability. Also, we preserve 2-minute of UPS for emergency situations, as we have used UPS beyond its main operation [22]. It has been proved leaving a 2-minute is sufficient to ensure a high availability of up to five nines (>0.99999770) [22].

In this section, we consider batteries in selecting the host in job allocation algorithm. Therefore, Algorithm 3 selects the appropriate host for a job based on the best-fit algorithm and the available charge of hosts' batteries.

Total energy cost for the proposed algorithm with introduced scenarios is shown in Table 6. As we can see from this table, using centralized and distributed UPS system, along with the proposed algorithm, decreases the energy cost 4.4% and 1.6%, respectively.

## 4 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a three-step algorithm which schedules heterogeneous workloads in a green data center, powered by solar energy and grid, cooled by free cooling and mechanical chiller system, and uses distributed UPSes. The complexity analysis of the proposed algorithm and its basic operations, exhibits that the proposed algorithm is a low complexity algorithm with a linear relation with the number of jobs, which makes it preferential to available optimization solutions. Moreover, presented results show that our proposed load management solution is feasible to electrical rate structure, energy prices, geographical characteristics, and load distributions. In the last section of the experimental results, we investigated using UPS system beyond its original purpose in order to reduce energy cost.

A future work would be the study of the effects of this usage on the batteries lifecycle, total energy costs, and energy leakage as well proposing solutions to manage UPS subsystem alongside workloads management.

## REFERENCES


[1] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *ACM SIGCOMM computer communication review*, vol. 39, no. 4. ACM, 2009, pp. 123-134.

[2] J. Hamilton, "Cooperative expendable micro-slice servers (cems): low cost, low power servers for internet-scale services," in *Conference on Innovative Data Systems Research (CIDR09) (January 2009)*. Citeseer, 2009.

[3] (2016) Google renewable energy. [Online]. Available: http://www.google.com/green/energy/

[4] (2016) Apple renewable energy. [Online]. Available: http://www.apple. com/environment/

[5] J. Ni and X. Bai, "A review of air conditioning energy performance in data centers," *Renewable and sustainable energy reviews*, vol. 67, pp. 625–640, 2017.

[6] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and abstracting total data center power," in *Workshop on Energy- Efficient Design*, vol. 11, 2009.

[7] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renewable and Sustainable Energy Reviews*, vol. 35, pp. 171–182, 2014.

[8] K.-P. Lee and H.-L. Chen, "Analysis of energy saving potential of air- side free cooling for data centers in worldwide climate zones," *Energy and Buildings*, vol. 64, pp. 103–112, 2013.

[9] D. Cheng, J. Rao, C. Jiang, and X. Zhou, "Elastic power-aware resource provisioning of heterogeneous workloads in self-sustainable datacenters," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 508–521, 2016.

[10] J. Li, Z. Bao, and Z. Li, "Modeling demand response capability by internet data centers processing batch computing jobs." *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 737–747, 2015.

[11] T.-W. Chang, C.-C. Lin, P. Liu, J.-J. Wu, C.-C. Shih, and C.-W. Huang, "Resource provision for batch and interactive workloads in data centers," in *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st Interna- tional Conference on*. IEEE, 2015, pp. 422–429.

[12] X. Lu, F. Kong, J. Yin, X. Liu, H. Yu, and G. Fan, "Geographical job scheduling in data centers with heterogeneous demands and servers," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Confer-*

*ence on*. IEEE, 2015, pp. 413–420.
[13] I˙. Goiri, M. E. Haque, K. Le, R. Beauchea, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, “Matching renewable energy supply and demand in green datacenters,” *Ad Hoc Networks*, vol. 25, pp. 520–534, 2015.
[14] S. Zhang, L. Pan, S. Liu, L. Wu, and X. Meng, “Profit based two-step job scheduling in clouds,” in *International Conference on Web-Age Information Management*. Springer, 2016, pp. 481–492.
[15] K. Haghshenas, A. Pahlevan, M. Zapater, S. Mohammadi, and D. Atienza, “Magnetic: Multi-agent machine learning-based approach for energy efficient dynamic consolidation in data centers,” *IEEE Transactions on Services Computing,* 2019.
[16] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing, “Utilizing green energy prediction to schedule mixed batch and service jobs in data centers,” in *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*. ACM, 2011, p. 5.
[17] Van Damme, Tobias, Claudio De Persis, and Pietro Tesi. "Optimized thermal-aware job scheduling and control of data centers." *IEEE Transactions on Control Systems Technology* 27.2 (2018): 760-771.
[18] J. Luo, L. Rao, and X. Liu, “Spatio-temporal load balancing for energy cost optimization in distributed internet data centers,” *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 387–397, 2015.
[19] L. Rao, X. Liu, L. Xie, and Z. Pang, “Hedging against uncertainty: A tale of internet data center operations under smart grid environment,” *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 555–563, 2011.
[20] (2016) Google server level ups batteries. [Online]. Available: http://www.datacenterknowledge.com/archives/2009/04/01/ efficient-ups-aids-googles-extreme-pue/
[21] (2016) Facebook server level ups batteries. [Online]. Available: http://www.datacenterknowledge.com/archives/2009/11/27/ facebook-follows-google-to-data-center-savings/
[22] S. Govindan, D. Wang, A. Sivasubramaniam, and B. Urgaonkar, “Aggressive datacenter power provisioning with batteries,” *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 1, p. 2, 2013.
[23] L. Yu, T. Jiang, Y. Cao, and Q. Qi, “Joint workload and battery scheduling with heterogeneous service delay guaranteesfor data center energy cost minimization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1937–1947, 2015.
[24] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, “Renewable and cooling aware workload management for sustainable data centers,” in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1. ACM, 2012, pp.175–186.
[25] L. Ganesh, H. Weatherspoon, T. Marian, and K. Birman, “Integrated approach to data center power management,” *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1086–1096, 2013.
[26] R. Zhou, Z. Wang, A. McReynolds, C. E. Bash, T. W. Christian, and R. Shih, “Optimization and control of cooling microgrids for data centers,” in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2012 13th IEEE Intersociety Conference on*. IEEE, 2012, pp. 338–343.
[27] F. P. Bleier, *Fan handbook*. McGraw-Hill, New York, 1998.
[28] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, “Making scheduling” cool”: Temperature-aware workload placement in data cen- ters.” in *USENIX annual technical conference, General Track*, 2005, pp. 61–75.
[29] M. vor dem Berge, G. Da Costa, A. Kopecki, A. Oleksiak, J.-M. Pierson, T. Piontek, E. Volk, and S. Wesner, “Modeling and simulation of data center energy-efficiency in coolemall,” in *International Workshop on Energy Efficient Data Centers*. Springer, 2012, pp. 25–36.
[30] E. Pakbaznia, M. Ghasemazar, and M. Pedram, “Temperature-aware dynamic resource provisioning in a power-optimized datacenter,” in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2010, pp. 124– 129.
[31] S. Ong, P. Denholm, and E. Doris, *The impacts of commercial electric utility rate structure elements on the economics of photovoltaic systems*. National Renewable Energy Laboratory, 2010.
[32] U. Pillai, “Drivers of cost reduction in solar photovoltaics,” *Energy Economics*, vol. 50, pp. 286–293, 2015.
[33] P. Olivella-Rosell, P. Lloret-Gallego, I. Munné-Collado, R. Villafafila- Robles, A. Sumper, S. Ø. Ottessen, J. Rajasekharan, and B. A. Bremdal, “Local flexibility market design for aggregators providing multiple flex- ibility services at distribution network level,” *Energies*, vol. 11, no. 4, p. 822, 2018.
[34] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, “Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems,” in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*. IEEE, 2010, pp. 1–9.
[35] (2017) Parallel workloads archive. [Online]. Available: http://www.cs. huji.ac.il/labs/parallel/workload
[36] (2017) The grid workloads archive. [Online]. Available: http://gwa.ewi.tudelft.nl
[37] *Power and Performance Data Sheet*, ENERGY STAR, 9 2008.
[38] X. Fan, W.-D. Weber, and L. A. Barroso, “Power provisioning for a warehouse-sized computer,” in *ACM SIGARCH computer architecture news*, vol. 35, no. 2. ACM, 2007, pp. 13–23.
[39] W. Wu, W. Wang, X. Fang, L. Junzhou, and A. V. Vasilakos, “Electricity price-aware consolidation algorithms for time-sensitive vm services in cloud systems*,” IEEE Transactions on Services Computing,* 2019.
[40] (2017) Forecast hourly. [Online]. Available: http://www.intellicast.com
[41] (2017) Pvwatts calculator. [Online]. Available: http://www.nrel.gov
[42] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine learning*, vol. 3, no. 2, pp. 95-99, 1988.
[43] (2017) Time of use pricing. [Online]. Available: https://www. portlandgeneral.com/residential/power-choices/ time-of-use/time-of-use-pricing
[44] (2017) Utility rates in Iran. [Online]. Available: http://www. howtoinvestiniran.com/utility-rates-in-iran/

**Kawsar Haghshenas** received the BS degree from K.N.Toosi University of Technology in 2012, and the MS degree from Sharif University of Technology in 2014, both in computer engineering. She is currently working toward the PhD degree at the Dependable Systems Design Laboratory, Electrical and Computer Engineering Department, University of Tehran. She was working in the Embedded Systems Laboratory at EPFL University from 2017 to 2018. Her research interests include VLSI circuits, energy optimization, cloud computing, and machine learning.

**Somayyeh Taheri** is a PHD student at the Energy Aware Systems Laboratory (EASY Lab) in the Department of Computer Engineering, at the Sharif University of Technology, Iran. She is completing her Ph.D. under the supervision of Prof. Maziar Goudarzi, with a thesis on Holistic Energy Management in Distributed Data Centers with Renewable Energy and Smart Grid. She completed her M.Sc. degree in 2012 at the same University. Her areas of interest include Cloud Computing, Distributed Data Center, and Energy Management. Under here Ph.D. studies, Somayyeh Taheri is actively involved in research on Efficient Utilization of Renewable Energy as well as Free Cooling in Data Center, Smart Power Management Unite (SPMU) in Data Center with Distributed UPS, Green Geographically Load Balancing (Green GLB), and Cost and Energy- Effective Big Data Processing.

**Maziar Goudarzi** (S’00, M’11, SM'16) is an Associate Professor at the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. He received the B.Sc., M.Sc., and Ph.D. degrees in Computer Engineering from Sharif University of Technology in 1996, 1998, and 2005, respectively. Before joining Sharif University of Technology as a faculty member in September 2009, he was a Research Associate Professor at Kyushu University, Japan from 2006 to 2008, and then a member of research staff at University College Cork, Ireland in 2009. His current research interests include architectures for large-scale computing systems, green computing, hardware software co-design, and reconfigurable computing. Dr. Goudarzi has won two best paper awards, published several papers in reputable conferences and journals, and served as member of technical program committees of a number of IEEE, ACM, and IFIP conferences including ICCD, ASP-DAC, ISQED, ASQED, EUC, and IEDEC among others.

**Siamak Mohammadi** received his BSc, MSc and PhD degrees from the University of Paris Sud Orsay, France in 1990, 1992 and 1996, respectively, all in electrical engineering. During his PhD he was supported by a grant from the Ministry of Education of France. From 1997 to 1999 he was a Research Associate with the Department of Computer Science, University of Manchester, England. In 1999 he moved to Canada and worked at Cogency Semiconductor Inc. in Toronto until 2003 and then at ATI Technologies Inc. until 2005. Currently he is an Associate Professor in School of Electrical and Computer engineering, at the University of Tehran, Iran. He has over 20 years' experience in VLSI digital area, ASIC design and verification, asynchronous design, and on-chip interconnects in GALS NoCs. He has contributed to the design of the asynchronous ARM microprocessor, as well as several PowerLine networking and RFID chips.