Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(3): 2403–2423.
DOI: 10.3934/mbe.2022111
Received: 28 September 2021
Revised: 13 December 2021
Accepted: 19 December 2021
Published: 05 January 2022

Research article

Bio-inspired negotiation approach for smart-grid colocation datacenter operation

Santiago Iturriaga*, **Jonathan Muraña and Sergio Nesmachnow**

Department of Computer Science, Universidad de la República, Julio Herrera y Reissig 565, Montevideo, Uruguay

* **Correspondence:** Email: siturria@fing.edu.uy; Tel: +59827142714.

Abstract: Demand response programs allow consumers to participate in the operation of a smart electric grid by reducing or shifting their energy consumption, helping to match energy consumption with power supply. This article presents a bio-inspired approach for addressing the problem of colocation datacenters participating in demand response programs in a smart grid. The proposed approach allows the datacenter to negotiate with its tenants by offering monetary rewards in order to meet a demand response event on short notice. The objective of the underlying optimization problem is twofold. The goal of the datacenter is to minimize its offered rewards while the goal of the tenants is to maximize their profit. A two-level hierarchy is proposed for modeling the problem. The upper-level hierarchy models the datacenter planning problem, and the lower-level hierarchy models the task scheduling problem of the tenants. To address these problems, two bio-inspired algorithms are designed and compared for the datacenter planning problem, and an efficient greedy scheduling heuristic is proposed for task scheduling problem of the tenants. Results show the proposed approach reports average improvements between 72.9% and 82.2% when compared to the business as usual approach.

Keywords: smart grid; demand response; colocation datacenter; evolutionary negotiation

1. Introduction

The smart grid paradigm is the current state-of-the-art for intelligent electricity networks [1]. It proposes the use of information and communication technologies for developing improved management and operation models that may account for planned demand growth, new energy sources, smart energy storage, and other relevant features of nowadays electric systems. Demand response is one of the main features of the smart grid paradigm. This feature enables customers to actively participate in the operation of the electric grid by reducing or rescheduling their energy consumption to match the power demand with the available power supply in exchange for some financial incentive. Demand response contributes to improving the economic operation and reliability of the electric system. Hence, these strategies are recognized as some of the most useful strategies in the active demand side theory [2].

Demand response relies on the flexibility of large energy consumers to defer some of their consumption in order to match an energy reduction target issued by the grid operator. This capability allows the grid operator to cope with energy demand and supply fluctuations by reshaping the demand curve of the whole electrical system. This enables the grid operator to flatten peaks of energy demand and deal with unexpected short-lived energy outages like the ones typically produced by renewable energy sources. Datacenters are a specific case of large consumers that can help the system by participating in demand response programs. This is because most datacenters can plan their activities beforehand by deferring the execution of some computational tasks and adjusting their energy consumption to contribute in demand response events [3].

A colocation datacenter is a widely used business model for datacenters in which the datacenter operator provides housing to third-party computing and networking equipment. In this model, each tenant rents ancillary services from the datacenter operator such as power, cooling and communication, and shares the datacenter facility with other tenants. The participation of a colocation datacenter in a demand response program presents a challenging problem since the datacenter operator does not control the actual computing equipment nor the execution of the computational tasks. Hence, the underlying control and planning problem must be modeled as a negotiation, focused on obtaining the best (i.e., most profitable) schedule for operation, considering the conflicting interests of different business actors such as the grid operator, datacenter operator and datacenter tenants.

In this line of work, this article presents an agent-based approach to address the problem of the smart operation of colocation datacenters to participate in the electricity market as active actors, with the role of providing services to meet demand response events. The problem is modeled considering a two-level hierarchy. On the one hand, the upper-level hierarchy addresses the datacenter planning problem modeling the interaction between the datacenter operator and its tenants. A Stackelberg evolutionary game is applied for modeling this interaction and two bio-inspired approaches are designed for minimizing the economic cost for the datacenter operator. On the other hand, the lower-level hierarchy addresses the tenant scheduling problem modeling the task execution schedule for each tenant. An efficient greedy scheduling heuristic is proposed for maximizing the economic profit of each tenant.

This article extends our previous work [4] by proposing a piece-wise demand response event. In this model, the time horizon of each demand response event is divided in several time intervals with different energy reduction targets. This provides the grid operator with a more adaptable and potent approach than our previous model. Furthermore, two mathematical formulations are proposed, one for each hierarchy, and two bio-inspired approaches to address the datacenter planning problem. The proposed bio-inspired approaches are validated and compared with each other over a realistic set of instances. The main results show that our proposed approach is effective and is able to reduce the economic cost for the datacenter operator by up to 82% when compared to a BaU approach.

This article is organized as follows. Next section presents a review of related works. The demand response planning problem for datacenters and its mathematical formulation are described in Section 3. The proposed bio-inspired approaches for addressing the datacenter planning problem are presented in Section 4 and the proposed efficient greedy scheduling heuristic for addressing the tenant scheduling problem is presented in Section 5. The experimental analysis is presented in Section 6 and conclusions and future work are presented in Section 7.

2. Related work

Game theory and Stackelberg games have been proposed to model negotiation approaches under the smart grid paradigm. Meng and Zeng [5] applied a Stackelberg game to model the interaction between an electricity retailer company and its residential customers. The goal of the electricity company is determining an appropriate pricing scheme in real time to maximize the profit. Customers are able to manage their energy consumption by planning the use of household appliances. The leader of the Stackelberg game is the electricity company, which applies a genetic algorithm for profit maximization, whereas for customers a linear programming model is applied for optimizing (i.e., minimizing) the electricity costs. Experimental results were reported for a neighbourhood with 1000 residential customers, each one having eight electric appliances, served by one electricity provider. Appliances were modeled using three categories: shiftable (i.e., admit planning), non-shiftable (i.e., do not admit planning), and curtailable (i.e., their consumption level can be reduced). Real time prices from Illinois power company from March 2012 were used. Results demonstrated that customers applying the proposed smart grid planning schemata were able to reduce the cost of the energy bill, when compared to users applying a BaU operation. These results were confirmed for different energy consumption patterns. Alshehri et al. [6] applied a non-cooperative Stackelberg game to model a scenario of multi-period demand response management in smart grid, considering several competitive agents. Authors proved that the proposed game has a unique Stackelberg equilibrium, where agents set prices to maximize profit and users accept prices to maximize benefits. In turn, the multi-period approach is more attractive than the single one, for users to participate in the game. Numerical computations were reported for different scenarios, modeling power availability for each agent, and results confirmed the previous findings about the convenience of the multi-period approach.

Yu and Hong [7] applied the Stackelberg game model for demand response in smart grids. The model considered one electricity company and several users. The main goal of the problem was balancing supply and demand, while smoothing the aggregated load in the electricity system. Optimal strategies within the proposed game are devised by solving optimization problems for each agent. The approach includes a theoretical pricing function for the regulation of real time prices. In turn, the applied pricing function is supposed to encourage users to participate in the game. The Stackelberg equilibrium is computed applying an iterative algorithm started by the electricity company. The algorithm is based on communicating proposed prices and updating their values according to the demands determined by users. Users are sequentially polled at each iteration, thus the computed prices depends on the order considered for polling. The iteration ends when all users converge to the same strategy (the equilibrium was achieved). The model was evaluated through simulations for a small case study considering one electricity company and three users, and a planning period of 24 hours divided in 24 time slots. Different generation costs were considered. Results showed that the proposed algorithm was able to adapt the demand, flatten peak consumption, and shifting load to valleys, thus properly matching electricity supply and demand. Scalability analysis for scenarios with 20 to 200 users were performed.

Dai et al. [8] studied real time pricing schemes for demand response to reduce both user and grid operating costs. A non-competitive Stackelberg game approach was proposed considering several retailers and multiple residential users. The coordination problem among residential users was modeled as an evolutionary game considering private information. The Lyapunov method was applied to find the evolutionary equilibrium of the replicator dynamics (i.e., convergence to a state where selections do not modify the population). In turn, a distributed algorithm was proposed to find the corresponding equilibrium of the Stackelberg game, assuming that retailers do not know any information about each other. The proposed model and algorithms were evaluated on a scenario considering two retailers and five residential users, and planning on a period divided into 24 time slots. Numerical results showed the convergence of both proposed algorithms for the case study. Furthermore, the proposed demand response model was effective when compared with a fixed pricing scheme, allowing residential users to benefit from reduced costs, thus encourage them to participate in the demand response program. No further scenarios were considered.

Regarding demand response for datacenters, Wang et al. [9] proposed a model to shift the computation loads to take advantage of either cheaper electricity or available renewable sources, under the cloud computing paradigm. A sequential Stackelberg game formulation was introduced to model the interaction between the smart grid operator and the datacenter operator considering two pricing schemes (real time and time ahead). The datacenter operator performs resource allocation over several distributed locations to maximize profit, whereas the smart grid operator aims at improving profit and guaranteeing a proper load balancing. For the optimization under power-dependant pricing, an iterative heuristic procedure was proposed, combining a resource allocation phase and a request dispatch phase. Both phases involve linear programming problems that can be solved within polynomial time complexity and are iteratively solved until reaching convergence. For the time-ahead pricing variant, a sub-optimal solution was computed by a two-step optimization method using heuristics and a Simulated Annealing metaheuristic to compute the price vectors. The experimental evaluation considered two systems with four and six datacenters. Results showed that the Stackelberg game model was effective to improve the profit and to reduce the risk of overflow in the power system.

In the work by Chen et al. [10], the datacenter operator negotiates with tenants the reduction of their energy consumption while tenants aim at maximizing their profits. An on-site generator is available for using when it is not possible to meet the target reduction with the reduction from tenants alone. The strategy used by the datacenter for negotiating with tenants consists in an iterative evaluation of a supply function mechanism, where the same monetary incentive per energy unit that is reduced is offered to all tenants equally. In each step of the iteration, the price is adjusted according to the tenants reduction until the target reduction is covered. The authors modeled the tenant monetary penalty function using a queuing theory model and considered a single reduction interval for the planning horizon. The minimization objective is the cost of using the on-site generator plus the monetary penalty incurred by the tenants in order to meet the energy reduction target. The authors proved that under convexity assumption of the tenant cost function, and assumptions about the cost of using the on-site generator, the proposed mechanism achieves the optimal solution. A case study of one datacenter with three tenants is presented. Results showed that the mechanism achieves solutions close to optimal.

Nguyen et al. [11] studied the demand response problem in colocation datacenters under the model of incentive tenants to reduce their electricity consumption. The interactions were modeled as a Stackelberg game with separated stages: finding an appropriate compensation cost by the demand response provider, determining the best strategy for each datacenter operator, and optimizing the energy reductions by tenants. For the first stage, efficient heuristics such as bisection and branch-and-bound were proposed and a theoretical quadratic cost function was considered to model the use of other energy generation sources. For the second stage, exact and approximate methods were proposed, based on algebraic formulations and a sequential optimization approach to find the optimum of the corresponding Stackelberg game. Finally, tenants reductions were computed according to a theoretical supply function that takes into account the energy surplus.

Chi et al. [12] studied an auction mechanism between a datacenter operator and its tenants and proposed a mixed-integer nonlinear programming problem for minimizing the social wellfare cost of the energy consumption. The authors propose to follow the Vickrey-Clarke-Groves reverse auction mechanism to guarantee the truthfulness of the proposal. According to this mechanism, the datacenter operator is the buyer and its tenants are the sellers, allowing each tenant to bid just one offer. The formulation considers an auxiliary on-site energy storage system and the energy consumption of the cooling system of the datacenter. The main variables of the proposed optimization problem correspond to the selection or rejection of each tenant offer and the amount of energy used from the on-site energy storage. The authors designed an approximation algorithm for addressing the problem. The proposed model was evaluated using real-world workloads and homogeneous servers, and its performance was compared with two approaches recently proposed by Zhang et al. [13] and Chen et al. [14]. The mechanism proposed by Chi et al. outperformed these two approaches, reducing the social welfare cost of the energy consumption by up to 28.34%.

Celik et al. [15] proposed a GA for energy management of a datacenter, that allows the participation in a time-of-use plan. Unlike emergency demand response events, where the start of the reduction intervals is known at short notice, Time-of-use plans have fixed reduction intervals in the year. However, similar strategies can be applied to reduce or delay energy consumption. The optimization problem minimizes the execution cost of the workload for the datacenter considering a scheduling horizon with high and low energy cost intervals. The authors consider workloads that include deferrable and non-deferrable tasks. Non-deferrable tasks are not schedulable. Hence, the optimization problem consists in scheduling the deferrable tasks. The energy consumption model for each server considers CPU usage alone and the cooling system is not modeled. Time-varying energy prices from two utility companies and workloads were generated for the experimental evaluation using traces from real cloud datacenters. Authors archived up to 11.58% cost reduction when compared to BaU.

Our previous articles [4, 16] proposed a two-level planning approach for the participation of colocation datacenters in demand response programs. Specifically, our previous research contribute by defining a market mechanism for the active participation of tenants, integrating a realistic function for computing the energy consumption of modern computing servers, and defining a simple thermal model for taking into account the energy consumed by the coolying system of the datacenter. This work extends our previous work by considering a piece-wise demand response event with multiple energy reduction targets. This provides the grid operator with a more adaptable and potent approach than our previous model. On top of that, a mathematical formulation is proposed for each level of the hierarchy and two bio-inspired metaheursitics algorithms are designed for guiding the optimization process.

To the best of our knowledge, this work includes several contributions to the state of the art. Regarding the datacenter modeling, a realistic model is incorporated to this work to evaluate the scheduling strategies of the tenants, considering the datacenter simulator proposed by Muraña and Nesmachnow [17]. On top of this, we incorporate the realistic energy consumption model introduced by Muraña et al. [18] to estimate the energy consumption of the computing servers. This model differentiates from the models presented in the related works by considering both CPU- and memory-intensive tasks in order to compute an accurate energy consumption prediction. Finally, the energy consumption of the cooling system is also taken into consideration unlike similar approaches proposed by Chen et al. [10] and Celik et al. [15]. These characteristics enable our proposed datacenter model to accurately simulate a real-world scenario.

Regarding the formulation of the optimization problem, multiple reduction intervals in the scheduling horizon are introduced in this work. This is a novel approach, since none of the reviewed works consider this characteristic for emergency demand-response programs. Considering multiple reduction intervals empowers the grid operator by enabling a fine-grainer reshaping of the energy demand curve. Furthermore, we propose an open bid mechanism with several bidding rounds, enabling multiple opportunities for tenants to change their energy reduction offer and compete with each other, potentially lowering the reduction cost for the datacenter. This contrasts with simpler mechanisms like the one proposed by Chi et al. [12], where tenants only bid once, and the operator selects the best offers in just one bidding round.

Finally, regarding the algorithmic approach, this work proposes two efficient upper-level bio-inspired metaheursitcis, one based on a genetic algorithm and the other based on particle swarm optimization (PSO). This approach proved to be accurate wihout requiring any assumption about the convenxity of the tenant cost function like the one proposed by Chen et al. [10].

3. Problem description and formulation

This section introduces the general problem description and presents the detailed problem formulation for each hierarchy in our proposed model.

3.1. Problem description

The proposed problem addresses the operation planning of a colocation datacenter participating in a demand response program. Colocation datacenters house servers and other IT equipment from different owners that are simply co-located in the same datacenter. That is, owners of IT equipment rent housing space and infrastructure services from the datacenter, such as cooling, connectivity and uninterruptible power supply. From here on, these owners will be referred as *tenants*.

This work considers the datacenter to be participating in an emergency demand response program. In such program, participants must cope with unplanned demand response events on short notice in order to address issues such as inaccurate generation forecast, extreme weather and problems in generation stations, among others. Events are notified as little as 10 minutes ahead and may span from a few minutes up to a few hours. Each event consists of several reduction intervals, each interval with an energy reduction target. Participants are obliged to fulfill the issued reduction target for each of the reduction intervals.

Colocation datacenters usually participate in emergency demand response programs by relying on in their on-site generators to reduce energy consumption from the grid and meet their energy reduction target [19]. This is considered to be the BaU strategy. However, fully relying on on-site generators is not cost-effective nor environmentally friendly.

This work considers an alternative strategy in which the datacenter operator offers monetary rewards to their tenants for them to willingly reduce their energy consumption during a demand response event. This strategy introduces the problem of how to effectively and efficiently negotiate with a set of tenants to reduce their energy consumption while minimizing the offered reward. Figure 1 presents the overall schema of the considered negotiation strategy.

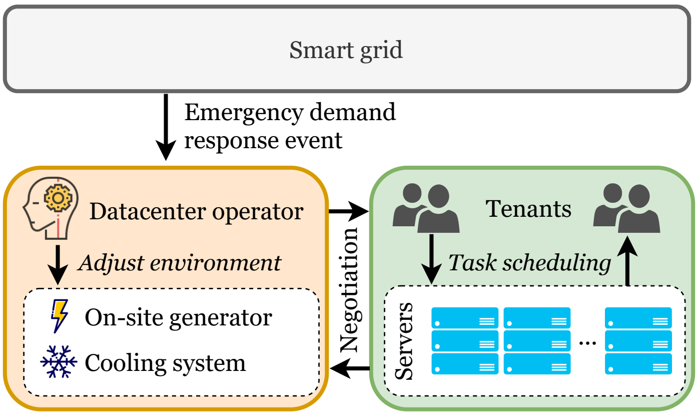


Figure 1. Schema of the proposed model for demand response planning in colocation datacenters.

3.2. *Problem formulation*

This section presents the formulation for the datacenter planning problem and the tenant scheduling problem. Both formulations consider a discrete-time and rolling-horizon approach.

3.2.1. Datacenter planning problem

The goal of this problem is to plan the operation of a colocation datacenter to minimize its budget in order to meet a demand response event in a smart grid environment. The proposed formulation considers the monetary incentive offered to tenants, the cost of operating the cooling system, and the cost of using the on-site diesel generator. The proposed formulation is as follows:

- T, a set of time steps $t \in T$ in the scheduling horizon.
- I, a collection of time intervals such that $\bigcup I \subseteq T$ and $\bigcap I = \emptyset$.
- α, the energy that must be reduced when compared to BaU operation by request of the electric market. This reduction must be attained at every time step of each interval.
- Let d_i be the power generated by the on-site diesel generator during time interval $I_i \in I$, $\bar{D}$ its maximum energy generation capacity, and γ the monetary cost per unit of energy per time step of using the generator.
- C, a set of tenants and $0 \leq r_i^c \leq \gamma \times |I_i|$ a monetary incentive offered to tenant $c \in C$ for each energy unit reduced on interval $I_i \in I$.
- $f : C \times I \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ models the energy a tenant is willing to reduce when compared to BaU on a time interval given a certain monetary incentive. Function f models the tenant scheduling problem, described in the next subsection.
- $g : \mathbb{R}^+ \times I \rightarrow \mathbb{R}^+$ models the budget saved by the cooling system given the energy consumption reduced by a tenant on a time interval.

The objective function is defined in Eqs (1)–(3).

$$\min \sum_{I_i \in I} \sum_{c \in C} \left(r_i^c - g(f(c, I_i, r_i^c), I_i) \right) + \sum_{I_i \in I} \left(d_i \times |I_i| \times \gamma \right) \tag{1}$$

Subject to:

$$\alpha \leq \ d_i + \sum_{c \in C} f(c, I_i, r_i^c) \ \forall I_i \in I \tag{2}$$

$$d_i \leq \ \bar{D} \ \forall I_i \in I \tag{3}$$

The objective presented in Eq (1) is to minimize the monetary cost for the datacenter operator while meeting the energy reduction target. The first term in Eq (1) corresponds to the reward offered to tenants minus the budget saved by the cooling system due to tenants usage reduction. The second term in Eq (1) corresponds to the cost of using the on-site diesel generator to further reduce the energy consumed from the grid to meet the reduction target. Constraint Eq (2) states the total energy reduction must be at least α for all time interval $I_i \in I$. Finally, constraint Eq (3) states the energy generated by the on-site generator must not exceed its maximum capability.

3.2.2. Tenant scheduling problem

In order to address the proposed datacenter planning problem, each tenant must address its own scheduling problem. The tenant scheduling problem is an underlying problem which consists in minimizing the energy consumption of a tenant given a certain monetary incentive offered by the datacenter operator.

The tenant scheduling problem is defined considering the following elements:

- T, a set of time steps $t \in T$ in the scheduling horizon.
- I, a collection of time intervals such that $\bigcup I \subseteq T$ and $\bigcap I = \emptyset$.
- $\hat{r}_i$, a monetary incentive offered to the tenant by the datacenter operator for each energy unit reduced on each interval $I_i \in I$.
- W, a workload (or set) of sequential and uninterruptible tasks. Each task $w \in W$ is described by: number of computing operations required for its completion w_l (i.e. its *length*), arrival time $w_a \in T$, due date $w_d \in T$ and the monetary penalty the tenant must pay if the task due date is not met w_p.
- S, a set of homogeneous multi-core servers comprised of $\hat{c}$ processing cores. Each processing core is capable of performing $\hat{o}$ computing operations each time step and is able to execute one task at a time.
- $l : S \times T \rightarrow \mathbb{N}$, indicates the number of tasks $w \in W$ executing in server $s \in S$ at time step $t \in T$.
- $m : S \times T \rightarrow \mathbb{R}^+$, indicates the energy reduced by server $s \in S$ at time step $t \in T$ when compared to BaU.
- $h^{'} : W \rightarrow S$ and $h^{''} : W \rightarrow T$, are the scheduling functions that determine the execution server $s \in S$ and starting time $t \in T$ of each task $w \in W$.
- $v : W \rightarrow \{0, 1\}$, the due date violation function such that $v(w \in W) = 0$ iif $h^{''}(w) + \frac{w_l}{\hat{o}} \leq w_d$. Otherwise, its value is 1.

The objective of the tenant scheduling problem is to maximize the profit of the tenant Eqs (4) and (5).

$$\max \sum_{I_i \in I} \hat{r}_i \times \Big(\min_{t \in I_i} \sum_{s \in S} m(s,t) \Big) - \sum_{w \in W} w_p \times v(w) \tag{4}$$

Subject to:

$$\hat{c} \geq \; l(s,t) \; \forall s \in S \;\; \forall t \in T \tag{5}$$

Equation (4) proposes maximizing the monetary profit of the tenant. The first term in Eq (4) corresponds to the profit earned by the tenant for reducing its energy consumption. The second term in Eq (4) corresponds to the monetary penalties paid by the tenant because of due dates not met.

4. Bio-inspired algorithms for datacenter planning on the smart grid

This section describes the proposed bio-inspired approach to plan the operation of a datacenter for participating in a demand response program.

4.1. Overall description of the resolution methodology

This work compares two bio-inspired algorithmic models to solve the datacenter planning problem. Both approach apply game theory to model the interactions between the datacenter operator and tenants and evolutionary computation to solve the optimization problem related to the negotiation.

The main concepts of game theory have been proved to explain the evolution of biological interactions [20]. In turn, evolutionary game theory is a branch of evolutionary theory particularly applicable for studying interactions between agents. Evolutionary game theory applies concepts of Darwinian evolution, competition based on natural selection, and transmission of hereditary characters between related individuals.

In the proposed resolution methodology, the overall negotiation scheme follows the Stackelberg game approach. The Stackelberg game, also referred as Stackelberg leadership model, is a market model from economics, applicable when a distinguished agent (the *leader*) is in such a position within the business model that allows him to perform an initial offer or move, and the rest of agents (the *followers*) react accordingly [21].

An equilibrium is achieved in the game (i.e., the market), when all agents agree on defining the best strategy for themselves, considering the strategies of the other agents. Thus, every player reaches a Nash equilibrium for the corresponding subgame between him and the leader. The model assumes perfect information for all agents and the profit maximization of each follower is based on the fact that the selected strategy do not affect the decisions of other followers. In those scenarios where the leader is firmly established, an equilibrium situation can be achieved after the negotiation lapse, in which all agents seek their profit maximization. In the considered Stackelberg game, the datacenter operator is the leader, since its actions guide the negotiation procedure, by defining proper incentives for tenants to reduce their power consumption. No altruism or social behaviour is considered, the behavior of agents is guided by selfishness, as the main criterion for strategic decisions. No strategy or procedure that does not tend to self-benefit is applied.

4.2. Bio-inspired methods for planning

Two bio-inspired metaheuristics are proposed for addressing the datacenter planning problem. Both are based on well-known metaheuristics. The first one is based on evolutionary algorithms (EAs) and the other on PSO. This section introduces some fundamental concepts about EAs and PSO, and presents the proposed planning algorithms.

4.2.1. Evolutionary algorithms

EAs are a set of non-deterministic optimization methods that emulate the evolutionary process of species in nature. The application of evolutionary approaches and self-replication for problem solving was suggested by pioneers in computer science in the decade of 1960, and the methodology was formalized and popularized between 1980 and 1990 [22].

In the last 30 years, EAs have been largely applied by the research community to solve complex real-world design and optimization problems in many areas. The most popular variant of EA are genetic algorithms (GA).

A GA is an iterative stochastic technique (according to the analogy with natural evolution, each iteration is called a generation). The optimization procedure works by applying stochastic operators on a set of candidate solutions (the population), in order to improve the fitness of solutions, a metric that evaluates how good each solution is to solve the considered problem, related to the function to optimize. The initial population is generated by either a randomized approach or by seeding the population using a randomized heuristic procedure or including problem-dependent knowledge. Different variation operators are applied to generate new candidate solutions. The canonical GA includes as main variation operators the recombination of solutions and random changes (mutations) in their contents, which are applied for building new solutions during the iterative cycle. The procedure is guided by a selection-of-the-best technique to tentative solutions of higher quality along the generations. The exploration of new solutions is iteratively applied until reaching a predefined stopping criterion, usually a fixed number of generation or a time limit. The following characteristics are considered of the proposed GA.

Solution representation In memory candidate solutions are represented as a vector $\vec{x} \in \mathbb{R}^{|C|\times|I|}$. Vector $\vec{x}$ represents the monetary incentive offered by the datacenter operator to each tenant on each time interval for each unit of energy reduced. That is, $x_1 = r_1^{c_1}$ is the incentive offered to $c_1 \in C$ during $I_1 \in I$, $x_2 = r_2^{c_1}$ is the incentive offered to $c_1 \in C$ during $I_2 \in I$, and so on.

Variation operators Well-known variation operators are applied to guide the evolution of the population. For the recombination of solutions the SBX crossover operator [23] is applied, while for the mutation operator the Polynomial mutation operator [24] is applied. These operators are randomly applied to solution selected from the population using the Binary tournament selection operator [25].

4.2.2. Particle swarm optimization

PSO is a computational method that mimics the movement in group of animals (e.g., a flock of birds, a swarm of insects, a shoal of fish, etc.) [26]. PSO was originally proposed for simulating social behavior in swarms, which has also been proved to model the movement of cells in the human body and other relevant natural phenomena. The main concept behind PSO is modeling and taking advantage of swarm intelligence, i.e., the collective behavior of decentralized, self-organized systems [27]. The collective behavior of agents interacting with one another and also with the environment, allows building a powerful global intelligence approach for solving complex problems.

PSO is an iterative technique intended to improve a set of candidate solutions (the *swarm*). The quality of each solution (*particle*) is evaluated by a given function. Particles are iteratively modified by movement operators that allows exploring the neighborhood of the search space. Movement operators are defined by simple mathematical formulas defined over the particle attributes (*position* and *velocity*). The movement of particles is also affected by already computed best positions for other particles in the swarm. The defined procedure guides the swarm towards the exploration and exploitation of better solutions in the search space.

In memory candidate solutions for PSO are represented exactly as for GA, i.e., a vector $\vec{x} \in \mathbb{R}^{|C|\times|I|}$ representing the monetary incentives offered to tenants. The variation operators of the PSO implemented in this article follows the standard PSO 2011 implementation by Zambrano-Bigiarini et al. [28].

Formally, given a D-dimensional search space, the position and velocity of particle $i = 1, 2, ..., N$ are defined as $\vec{X_i} = \{x_{i1}, x_{i2}, ..., x_{iD}\}$ and $\vec{V_i} = \{v_{i1}, v_{i2}, ..., v_{iD}\}$, respectively. On top of this, the best previous position of particle i and the best found position for the neighborhood of each particle are defined as $\vec{P_i} = \{p_{i1}, p_{i2}, ..., p_{iD}\}$ and $\vec{L} = \{l_1, l_2, ..., l_D\}$. Standard PSO 2011 proposes to compute a center of gravity ($\vec{G_i}$), as shown in Eq (8), where $t = 1, 2, ..., T$ are the iteration steps of the PSO algorithm, $\otimes$ is the element-wise multiplication of vectors, c_1 is the cognitive coefficient, c_2 is the social coefficient, and $\vec{U}_1^t$ and $\vec{U}_2^t$ are random uniformly distributed vectors in $[0, 1]^D$-space [29].

$$\vec{p}_i^t = \vec{X}_i^t + c_1 \vec{U}_1^t \otimes \left(\vec{P}_i^t - \vec{X}_i^t\right) \tag{6}$$

$$\vec{l}_i^t = \vec{X}_i^t + c_2 \vec{U}_2^t \otimes \left(\vec{L}^t - \vec{X}_i^t\right) \tag{7}$$

$$\vec{G}_i^t = \frac{\vec{X}_i^t + \vec{p}_i^t + \vec{l}_i^t}{3} \tag{8}$$

Finally, velocity and position of particle i are updated as shown in Eqs (9) and (10), where $\mathcal{H}_i\left(\vec{G}_i^t, ||\vec{G}_i^t - \vec{X}_i^t||\right)$ is a hyperspherical distribution with center $\vec{G}_i^t$ and radius $||\vec{G}_i^t - \vec{X}_i^t||$.

$$\vec{V}_i^{t+1} = \omega \vec{V}_i^t + \mathcal{H}_i\left(\vec{G}_i^t, ||\vec{G}_i^t - \vec{X}_i^t||\right) - \vec{X}_i^t \tag{9}$$

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1} \tag{10}$$

The resulting search, using the center of gravity for velocity and position update, is less biased than the standard search and does not depend on the system of coordinates used. Thus, the approach is suitable for solving complex optimization problems such as the one addressed in this article, especially when no information about the fitness landscape of the optimization problem is available.

Regarding parameters, the proposed implementation considers an adaptive random topology of size K applied for computing $\vec{L}$, the inertia weight ω is set to $1/(2 \times ln(2))$, and both c_1 and c_2 are set to $1/2 + ln(2)$, following suggestions by Zambrano-Bigiarini et al. [28]. Furthermore, different values for parameters K and N are studied in the experimental evaluation reported in Section 6.4.

5. Efficient greedy heuristic for task scheduling for each tenant

The proposed heuristic follows a dynamic scaling approach for smart energy management [30, 31] and works iteratively by constraining the maximum relative number of computing cores that are usable on each interval for a given tenant. The rationale of this is that constraining the number of usable computing cores may reduce energy consumption, depending on the computing demand of the scheduled tasks. From now on, the maximum relative number of usable computing cores on each interval will be referred as the processing level of each interval. It is defined by vector $\hat{p}$, with $\hat{p}_i$ being the processing level of interval I_i. For example, lets consider a scenario where $|I| = 3$ and a tenant with $\hat{p} = (50, 30, 100)$. Then at most 50% of the computing cores of the tenant will be used during interval I_1, at most 30% will be used during interval I_2, and all the cores may be used during interval I_3. This means that 50% of the computing cores will be idle during interval I_1 and 70% will be idle during interval I_2.

The proposed heuristic optimizes one interval at a time aiming to find the processing level that maximizes the total profit Eq (4). Initially, a vector $\hat{p}$ is constructed such that $\hat{p}_i = 100$ for all $I_i \in I$. On each iteration of the heuristic, the processing level of interval I_i is optimized by considering a pre-defined set of candidate processing levels, L. Hence, a total number of $|L|$ different candidate $\hat{p}'$ are constructed by setting different values for $\hat{p}'_i$ while the processing levels of the rest of the intervals are left unchanged, i.e., $\hat{p}'_j = \hat{p}_j \forall j \neq i$. After that, a simple list scheduling strategy is applied to evaluate each vector $\hat{p}'^{1..|L|}$ by computing a task-to-core assignment for all the intervals, subject to the processing level constraint of each vector. The list scheduling strategy sorts the queue of pending tasks (i.e. arrived but not yet executed) in order of decreasing monetary penalty, w_p. This way, tasks with the most monetary penalty for the tenant are prioritized. The energy consumption of the schedule is computed using the server energy model proposed in our previous work [18] taking into account the type of each task (i.e. CPU-bound or Memory-bound). Finally, the value $\hat{p}_i$ is updated with the most profitable value of $\hat{p}'^{1..|L|}$ for I_i. Next, the heuristic starts a new iteration for optimizing I_{i+1} and keeps iterating until there are no more intervals to optimize. The schema of the proposed heuristic is presented in Algorithm 1.

6. Experimental analysis

This section presents the set of realistic problem instances created for the experimental evaluation and the execution and development infrastructure. After that, the parameter calibration study and the experimental analysis are presented.

6.1. Problem instances

A total of ten realistic problem instances of different sizes were generated to evaluate the proposed planning algorithms. A scheduling horizon of 60 minutes is considered for every instance. The considered demand-response event consists of 3 energy reduction intervals of 10 minutes such that $I_1 = [5, 14]$, $I_2 = [25, 34]$ and $I_3 = [45, 54]$.

Table 1 shows the characteristics of the generated instances. One small-sized instance was generated for calibrating the proposed algorithms and three medium-, large- and huge-sized instances were generated for evaluating the proposed algorithms. For each instance, the number of tasks and number

Algorithm 1 Schema of the proposed greedy heuristic for task scheduling

```
max_profit ← $-\infty$
$\hat{p}$ ← [100, 100, 100, ..]
$\Gamma_{best}$ ← compute scheduling for processing level vector $\hat{p}$
for $I_i$ in $I$ do
    for $L_j$ in $L$ do
        $\hat{p}_i[I_i]$ ← $L_j$
        $\Gamma$ ← compute scheduling for processing level vector $\hat{p}$
        profit ← compute profit for schedule $\Gamma$
        if profit > max_profit then
            max_profit ← profit
            $\Gamma_{best}$ ← $\Gamma$
        end if
    end for
end for
$\hat{M}$ ← compute energy reduced by $\Gamma_{best}$
return $\hat{M}$
```

Table 1. Characteristics of the realistic problem instances generated for this work.

Size	# Tenants	# Tasks per tenant	# Servers per tenant	Energy reduction (kW)	$\|L\|$
Small	9	3500–5500	20–30	6	50
Medium	27	3500–5500	20–30	18	50
Large	18	7000–11000	40–60	36	50
Huge	54	7000–11000	40–60	108	10

of servers per tenant were chosen randomly on the specified range, following a uniform distribution. Column $|L|$ describes the size of the set L considered in the proposed greedy heuristic for task scheduling. Each processing level $l_i \in L$ is $l = i \times \frac{100}{|L|}$ with $i = 1, 2, ..|L|$

Servers are comprised of 24 computing cores, each capable of 3000 million of instructions per seconds (MIPS). Workloads of tasks for tenants were generated based on real workloads from the parallel workloads archive (PWA) [32]. These workloads were adapted to the model proposed in this work. To create a workload W for the proposed model the following procedure is preformed. First, a task w is randomly chosen from the PWA workload with uniform distribution and W is created considering the desired number of consecutive tasks from the PWA workload starting with w.

Arrival time for each task w is adjusted to the desired planning horizon, T. Let a_{start} and a_{end} be the PWA arrival time of the first and last tasks selected for W, then the arrival time of task $w \in W$ is computed as $w_a = \frac{(\hat{w}_a - a_{start})}{(a_{end} - a_{start})} \times |T|$, with $\hat{w}_a$ being the PWA arrival time of w. Length of task $w \in W$ is computed considering the CPU time (in seconds) of the task in the PWA workload, $\hat{w}_t$. Since processor speed is 1000 MIPS according to the PWA workload, the length of task w in millions of instructions is computed as $w_l = \hat{w}_t \times 1000$. Due date of task w is computed randomly as $w_d = \frac{w_l}{\hat{o}} \times (1 + R)$, where $R \in [0.1, 1]$ is a random value with uniform distribution.

In order to compute the energy consumption, each task is classified in a certain type: CPU-bound or memory-bound. This classification is computed according to the memory-usage information from the PWA workload. If the memory usage of the task is greater or equal to 200 MB, the task is considered to be memory-bound, otherwise is CPU-bound. The monetary penalty w_p of a task is chosen randomly with a Poisson distribution with $\lambda = 3$.

Finally, regarding the datacenter. This work considers a simple linear function based on the power usage effectiveness (PUE) [33] for modeling the cooling system. A PUE of 1.5 was considered for all instances regarding the cooling system of the datacenter. This is a reasonable value according recent studies [33]. Furthermore, the on-site diesel generator is considered to be capable to power the normal operation of the datacenter during the whole demand-response event. Hence, the datacenter is always capable to comply with the reduction target. The monetary cost per unit of energy per time step when using the generator is 0.4 \$/kWh for all instances. This cost considers realistic values for the price of diesel fuel and fuel consumption of a standard diesel generator [10,34].

6.2. Evaluation metrics

Several metrics are considered for evaluating the proposed approach. The proposed metrics take into consideration the business perspective of both the datacenter and its tenants. The considered metrics are the following:

- Datacenter cost improvement over BaU (ΔC) reports the relative improvement of the datacenter cost comparing to the BaU strategy when participating in a demand-response event. Let C be the best cost computed by the proposed method and C_{BaU} be the cost when applying BaU strategy, then $\Delta C = \frac{C - C_{BaU}}{C_{BaU}} \times 100$.
- Tenants violated tasks over BaU (ΔVT) reports the relative increase of violated tasks when comparing to the BaU strategy. A task is considered to be violated when it does not finish its execution before its due time, or it is not executed at all. Let W be the total number of tasks to be executed, VT the number of violated tasks when applying the proposed method and VT_{BaU} the number of violated tasks when applying the BaU strategy, then $\Delta VT = \frac{VT - VT_{BaU}}{W} \times 100$.
- Tenants non-executed tasks over BaU (ΔNT) reports the relative increase of non-executed tasks when comparing to the BaU strategy. A non-executed task is a violated task that is not executed at all. Let W be the total number of tasks to be executed, NT the number of non-executed tasks when applying the proposed method and NT_{BaU} the number of non-executed tasks when applying the BaU strategy, then $\Delta NT = \frac{NT - NT_{BaU}}{W} \times 100$.
- Tenants tardiness over BaU (ΔT) reports the relative increase of tardiness of tasks when comparing to the BaU strategy. Tardiness is a well-known metric defined as the $\max(0, CT - DT)$, where CT is the task completion time and DT is its due time. Let T be the sum of the tardiness of all executed tasks when the proposed method is applied, T_{BaU} be the sum of the tardiness of all executed tasks when the BaU strategy is applied, then $\Delta T = (T - T_{BaU})/T_{BaU}$.

6.3. Execution and development infrastructure

All the proposed algorithms were implemented in Java. A datacenter simulator, introduced in our previous work [17], is used to simulate the execution the tasks on the datacenter. The proposed metaheuristics were implemented using the jMetal framework [35], a Java-based framework for single- and multi-objective optimization with metaheuristics.

Experiments were executed in ClusterUY, a collaborative scientific HPC infrastructure in Uruguay. Detailed infrastructure information is available at https://cluster.uy.

6.4. *Parameter calibration study*

The goal of the calibration study is to evaluate the accuracy of each metaheuristic with different configuration of parameters in order to find the best configuration for each metaheuristic. A total of 9 configurations of parameters were studied for each metaheuristic. The parameter calibration study was performed by addressing one small problem instance. A set of 30 independent executions for each metaheuristic and for each combination of parameter values. The parameters evaluated for GA were $p_c = \{1.0, 0.9, 0.8\}$ for applying SBX crossover and $p_m = \{0.1, 0.05, 0.01\}$ for applying polynomial mutation. Population size was configured as $N_{ga} = 100$. These are commonly suggested values for the GA parameters [36, 37]. The parameters calibrated for PSO were swarm size $N_{pso} = \{50, 75, 100\}$ and neighborhood size $K = \{3, 5, 7\}$. These are the suggested values for Standard PSO 2011 [26, 28]. Every combination of the proposed parameter values was evaluated for each metaheuristic with a stopping condition of 25000 evaluations for every configuration. Figure 2a presents the datacenter cost computed by GA for every combination of parameters while Figure 2c presents the datacenter cost computed by PSO for every combination of parameters.

The Kruskal-Wallis H-test indicate results from GA and PSO do not originate from the same distribution with p-value ≤ 0.001. After that, the best configuration is selected by performing a post hoc analysis by applying the Dunn H-test for multiple comparisons of mean rank sums. Figure 2b shows the post hoc analysis results for GA and Figure 2d for PSO.

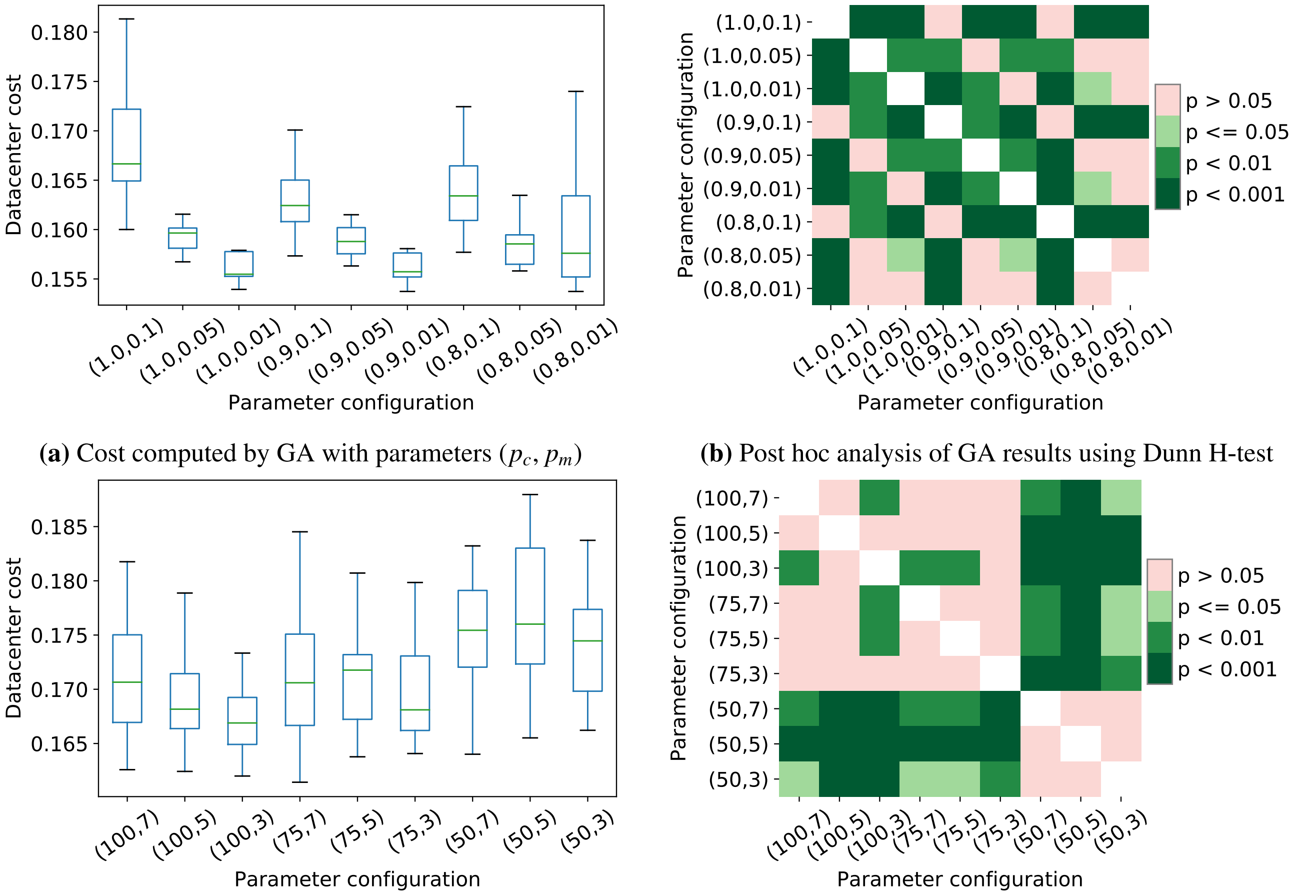
Post hoc analysis shows $(1.0, 0.01)$ and $(0.9, 0.01)$ are the best configurations for GA since both compute the lowest median cost and are significantly different from most of the other configurations. However, $(0.9, 0.01)$ was able to compute a slightly lower median cost than $(1.0, 0.01)$. PSO post hoc analysis clearly shows $(100, 3)$ is the best configuration. Hence, from here onwards, $(0.9, 0.01)$ configuration is used for GA and $(100, 3)$ for PSO.

6.5. *Main results and discussion*

A total of 30 independent executions were performed for every problem instance for each metaheuristic with the best configuration for GA and PSO. Figure 3 presents the datacenter cost computed by GA and PSO for every problem instance. Results show GA is more accurate in median than PSO when addressing most instances, disregarding their size.

Table 2 presents the median and IQR comparison of the computed datacenter cost. The Kruskal-Wallis H-test confirms GA is significantly more accurate than PSO for all instances. Overall, GA is 4.9% more accurate than PSO in average, and up to 9.3% more accurate than PSO.

The presented results show the proposed GA planning algorithm is significantly more accurate than PSO from the datacenter operator point of view. However, it is fundamental to study the whole impact of applying the proposed strategies from a business point of view. Hence, four different metrics are proposed to evaluate the impact of the proposed strategies over BaU: (i) datacenter cost improvement (ΔC), (ii) tenants violated tasks (ΔVT), (iii) tenants non-executed tasks (ΔNT) and (iv) tenants tardiness (ΔT). Table 3 presents the results of the proposed metrics for GA and PSO. Results show



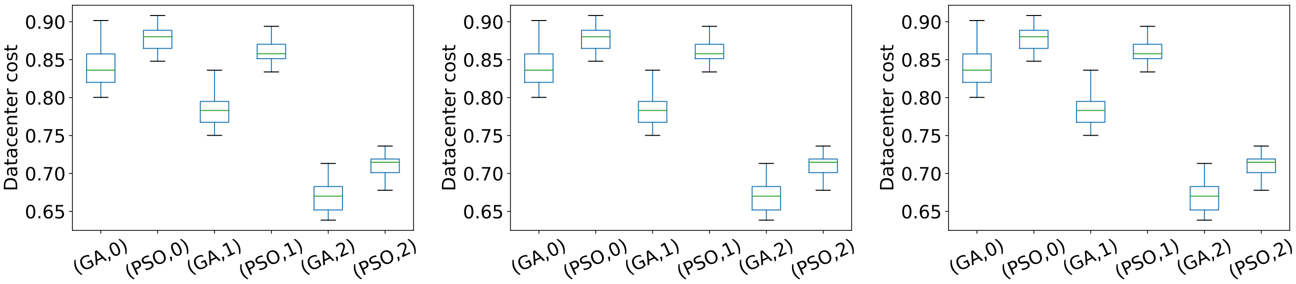
(a) Cost computed by GA with parameters (p_c, p_m)

(b) Post hoc analysis of GA results using Dunn H-test

(c) Cost computed by PSO with parameters (N_{pso}, K)

(d) Post hoc analysis of PSO results using Dunn H-test

Figure 2. Datacenter cost computed by GA ans PSO after 30 independent executions addressing a small-sized problem instance with every combination of configuration of parameters.



(a) Medium-sized problem instances

(b) Large-sized problem instances

(c) Huge-sized problem instances

Figure 3. Datacenter cost computed by GA and PSO after 30 independent executions for every problem instance referenced by (*planning algorithm*, *instance number*).

Table 2. Median and interquartile range (IQR) comparison of the datacenter cost computed by GA and PSO for every problem instance.

Instance	*GA*		*PSO*		p-value
number	*Median*	*IQR*	*Median*	*IQR*	
Medium-sized instances					
0	0.84	0.038	0.88	0.024	< 0.001
1	0.78	0.028	0.86	0.019	< 0.001
2	0.67	0.031	0.71	0.018	< 0.001
Large-sized instances					
0	1.97	0.082	2.05	0.037	< 0.001
1	1.86	0.047	1.94	0.049	< 0.001
2	1.64	0.071	1.70	0.041	< 0.001
Huge-sized instances					
0	5.41	0.139	5.68	0.141	< 0.001
1	5.60	0.097	5.86	0.137	< 0.001
2	5.77	0.132	6.02	0.149	< 0.001

GA and PSO improve datacenter cost consistently over BaU for all instances, with improvements of 76.7 ± 2.9% and 75.4 ± 2.8%, respectively.

For tenants, business impact varies depending on instance size. For medium-sized instances, the number of additional affected tasks over BaU is around 2.1 ± 1.0% for both GA and PSO. Affected tasks do not produce a significant impact on tardiness, with increases of only 1.04 ± 0.91 times and 1.11 ± 0.81 times over BaU for GA and PSO. Finally, considering the whole workload there are an additional 2.4 ± 1.1% of tasks that are not executed at all comparing to BaU when applying GA and PSO. These tasks are either postponed to the following planning horizon or are definitely cancelled. For large- and huge-sized instances, the impact is significantly higher than for medium-sized instances. The number of additional affected tasks over BaU is around 4.9 ± 0.6% for both GA and PSO. That is, more than twice than medium-sized instances. Furthermore, affected tasks produce a significant impact on tardiness, with increases of 6.0 ± 3.8 times and up to 13.7 times over BaU for GA and PSO. This introduces a significant delay on task execution with respect to BaU. Finally, considering the whole workload there are an additional 3.3 ± 0.9% of tasks that are not executed on the planned horizon when comparing to BaU.

7. Conclusions and future work

This work studied the problem of a colocation datacenter participating in a smart grid emergency demand-response program. The problem is divided hierarchically into two sub-problems, the *datacenter planning problem* and the underlying *tenant scheduling problem*. A realistic mathematical formulation is presented for both sub-problems, considering a discrete-time approach with a rolling

Table 3. Datacenter cost improvement (ΔC), tenants violated tasks (ΔVT), non-executed tasks (ΔNT) and tardiness (ΔT), over BaU for medium-, large- and huge-sized instances.

Instance	*GA*				*PSO*			
number	ΔC	ΔVT	ΔNT	ΔT	ΔC	ΔVT	ΔNT	ΔT
Medium-sized instances								
0	77.8%	2.8%	3.7%	2.01×	76.4%	2.8%	3.8%	1.97×
1	79.2%	2.7%	2.0%	0.22×	76.9%	2.7%	1.9%	0.35×
2	82.2%	1.0%	1.6%	0.88×	81.1%	1.0%	1.8%	0.97×
Large-sized instances								
0	73.2%	6.1%	1.9%	13.52×	72.2%	6.8%	1.8%	13.68×
1	75.0%	4.3%	3.1%	3.43×	73.8%	4.7%	3.8%	3.10×
2	78.2%	4.5%	4.7%	3.99×	77.2%	4.9%	4.3%	3.96×
Huge-sized instances								
0	75.9%	4.7%	2.7%	3.98×	74.6%	4.6%	3.2%	4.16×
1	74.7%	4.8%	3.8%	7.43×	73.6%	4.9%	3.5%	6.59×
2	74.3%	4.7%	3.7%	4.35×	72.9%	4.6%	3.3%	3.98×

horizon.

Approximate algorithms were proposed for addressing the formulated problems. On the one hand, two bio-inspired algorithms are proposed for addressing the datacenter planning problem. A GA that is based on the natural evolution of the species and a PSO algorithm that is based on the movement of a swarm of insects. Both algorithms consider the same in-memory representation for the problem but apply a different algorithmic approach for computing a near-optimal problem solution. On the other hand, an efficient greedy heuristic is proposed for addressing the tenant scheduling problem. The proposed greedy heuristic is based on a simple but effective iterative constructive approach.

A total of ten realistic synthetic problem instances ranging from small to huge size were created for evaluating the proposed algorithms. All problem instances were generated based on real tasks workloads from the PWA [32] and include instances with more than 378,000 tasks.

Experimental analysis shows that the best results are computed for medium-sized instances. GA and PSO achieve the highest improvement on datacenter cost while suffering the lowest impact on task execution when addressing medium-sized instances. Large- and huge-sized instances are also profitable for the datacenter. However, impact on task execution increases significantly when comparing to medium-sized instances and tenants are more reluctant to reduce their energy consumption. Overall, results show the effectiveness of the proposed approach, reporting average improvements of the datacenter cost over Business as Usual (BaU) of more than 75%.

The main lines of future work are the following. Construct additional realistic problem instances considering a wider spectrum of scenarios. Design a diverse set of lower-level heuristics for addressing the tenant scheduling problem, taking into account the diversity of goals and priorities tenants may have. Design improved upper-level bioinspired algorithms based on promising and recently proposed metaheuristics such as the flower pollination algorithm [38] and the Harris Hawks optimization algorithm [39]. And finally, formulate and propose an algorithmic approach for addressing a true multiobjective problem, simultaneously considering the goals of the datacenter operator and the goals of its tenants.

References

1. J. Momoh, *Smart grid: Fundamentals of design and analysis*, Wiley IEEE Press, 2012.
2. H. Fraser, The importance of an active demand side in the electricity industry, *Electr. J.*, **14** (2001), 52–73. https://doi.org/10.1016/S1040-6190(01)00249-4
3. M. Chen, C. Gao, M. Song, S. Chen, D. Li, Q. Liu, Internet data centers participating in demand response: A comprehensive review, *Renew. Sustain. Energy Rev.*, **117** (2020), 1–15. https://doi.org/10.1016/j.rser.2019.109466
4. J. Muraña, S. Nesmachnow, S. Iturriaga, S. M. de Oca, G. Belcredi, P. Monzón, et al., Two level demand response planning for retail multi-tenant datacenters, in *18th International Conference on High Performance Computing and Simulation*, (2021), 1–8.
5. F. L. Meng, X. J. Zeng, A Stackelberg game-theoretic approach to optimal real-time pricing for the smart grid, *Soft Comput.*, **17** (2013), 2365–2380. https://doi.org/10.1007/s00500-013-1092-9
6. K. Alshehri, J. Liu, X. Chen, T. Basar, A Stackelberg game for multi-period demand response management in the smart grid, in *54th IEEE Conference on Decision and Control*, (2015), 5889–5894. https://doi.org/10.1109/CDC.2015.7403145
7. M. Yu, S. Hong, Supply-demand balancing for power management in smart grid: A Stackelberg game approach, *Appl. Energy*, **164** (2016), 702–710. https://doi.org/10.1016/j.apenergy.2015.12.039
8. Y. Dai, Y. Gao, H. Gao, H. Zhu, Real-time pricing scheme based on Stackelberg game in smart grid with multiple power retailers, *Neurocomputing*, **260** (2017), 149–156. http://doi.org/10.1016/j.neucom.2017.04.027
9. Y. Wang, X. Lin, M. Pedram, A Stackelberg game-based optimization framework of the smart grid with distributed PV power generations and data centers, *IEEE Trans. Energy Conver.*, **29** (2014), 978–987. https://doi.org/10.1109/TEC.2014.2363048
10. N. Chen, X. Ren, S. Ren, A. Wierman, Greening multi-tenant data center demand response, *Perform. Eval.*, **91** (2015), 229–254. https://doi.org/10.1016/j.peva.2015.06.014
11. M. N. H. Nguyen, D. Kim, N. H. Tran, C. S. Hong, Multi-stage Stackelberg game approach for colocation datacenter demand response, in *19th Asia-Pacific Network Operations and Management Symposium*, (2017), 139–144. https://doi.org/10.1109/APNOMS.2017.8094193
12. C. Chi, F. Zhang, K. Ji, A. Marahatta, Z. Liu, Improving energy efficiency in colocation data centers for demand response, *Sustain. Comput. Infor. Syst.*, **29** (2021), 100476. https://doi.org/10.1016/j.suscom.2020.100476
13. L. Zhang, S. Ren, C. Wu, Z. Li, A truthful incentive mechanism for emergency demand response in colocation data centers, in *IEEE Conference on Computer Communications*, (2015), 2632–2640. https://doi.org/10.1109/INFOCOM.2015.7218654

14. J. Chen, D. Ye, S. Ji, Q. He, Y. Xiang, Z. Liu, A truthful FPTAS mechanism for emergency demand response in colocation data centers, in *IEEE Conference on Computer Communications*, (2019), 2557–2565. https://doi.org/10.1109/INFOCOM.2019.8737468

15. B. Celik, G. Rostirolla, S. Caux, P. Renaud-Goud, P. Stolf, Analysis of demand response for datacenter energy management using GA and time-of-use prices, in *IEEE PES Innovative Smart Grid Technologies Europe*, (2019), 1–5. https://doi.org/10.1109/ISGTEurope.2019.8905618

16. J. Muraña, S. Nesmachnow, S. Iturriaga, S. M. de Oca, G. Belcredi, P. Monzón, et al., Negotiation approach for the participation of datacenters and supercomputing facilities in smart electricity markets, *Program. Comput. Software*, **46** (2020), 636–651. https://doi.org/10.1134/S0361768820080150

17. J. Muraña, S. Nesmachnow, Simulation and evaluation of multicriteria planning heuristics for demand response in datacenters, *Simulation*, (2021), 1–18. https://doi.org/10.1177/00375497211020083

18. J. Muraña, S. Nesmachnow, F. Armenta, A. Tchernykh, Characterization, modeling and scheduling of power consumption of scientific computing applications in multicores, *Cluster Comput.*, **22** (2019), 839–859. https://doi.org/10.1007/s10586-018-2882-8

19. N. H. Tran, C. Pham, S. Ren, Z. Han, C. S. Hong, Coordinated power reduction in multi-tenant colocation datacenter: An emergency demand response study, in *IEEE International Conference on Communications*, (2016), 1–6. https://doi.org/10.1109/ICC.2016.7511560

20. C. Cowden, Game theory, evolutionary stable strategies and the evolution of biological interactions, *Nat. Educ. Knowl.*, **3** (2012), 1–6.

21. H. Stackelberg, *The theory of the market economy*, Oxford University Press, 1952.

22. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.

23. K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex syst.*, **9** (1995), 115–148.

24. K. Deb, S. Tiwari, Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization, *Eur. J. Oper. Res.*, **185** (2008), 1062–1087. https://doi.org/10.1016/j.ejor.2006.06.042

25. D. E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, *Found. Genet. Algorithms*, **1** (1991), 69–93, https://doi.org/10.1016/B978-0-08-050684-5.50008-2

26. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of International Conference on Neural Networks*, (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

27. G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in *Robots and Biological Systems: Towards a New Bionics?*, (1993), 703–712. https://doi.org/10.1007/978-3-642-58069-7_38

28. M. Zambrano-Bigiarini, M. Clerc, R. Rojas, Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements, in *IEEE Congress on Evolutionary Computation*, (2013), 2337–2344. https://doi.org/10.1109/CEC.2013.6557848

29. M. Clerc, *Particle swarm optimization*, John Wiley and Sons, 2010.

30. A. Gandhi, M. Harchol-Balter, R. Raghunatha, M. A. Kozuch, Autoscale: Dynamic, robust capacity management for multi-tier data centers, *ACM Trans. Comp. Sys.*, **30** (2012), 1–26. https://doi.org/10.1145/2382553.2382556

31. M. Lin, A. Wierman, L. L. Andrew, E. Thereska, Dynamic right-sizing for power-proportional data centers, *IEEE ACM Trans. Netw.*, **21** (2012), 1378–1391. https://doi.org/10.1109/INFCOM.2011.5934885

32. D. G. Feitelson, D. Tsafrir, D. Krakov, Experience with using the parallel workloads archive, *J. Parallel Distrib. Comput.*, **74** (2014), 2967–2982. https://doi.org/10.1016/j.jpdc.2014.06.013

33. L. A. Barroso, U. Hölzle, P. Ranganathan, The datacenter as a computer: Designing warehouse-scale machines, Morgan and Claypool Publishers LLC, 2018.

34. V. Oladokun, O. Asemota, Unit cost of electricity in Nigeria: A cost model for captive diesel powered generating system, *Renew. Sustain. Energy Rev.*, **52** (2015), 35–40. https://doi.org/10.1016/j.rser.2015.07.028

35. J. Durillo, A. Nebro, jMetal: A Java framework for multi-objective optimization, *Adv. Eng. Software*, **42** (2011), 760–771. https://doi.org/10.1016/j.advengsoft.2011.05.014

36. A. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, *IEEE Trans. Evol. Comput.*, **3** (1999), 124–141. https://doi.org/10.1109/4235.771166

37. A. E. Eiben, S. K. Smit, *Evolutionary Algorithm Parameters and Methods to Tune Them*, Springer, (2012), 15–36.

38. X. S. Yang, Flower pollination algorithm for global optimization, in *Unconventional Computation and Natural Computation*, (2012), 240–249. https://doi.org/10.1007/978-3-642-32894-7_27

39. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028