SHIELD: Sustainable Hybrid Evolutionary Learning Framework for Carbon, Wastewater, and Energy-Aware Data Center Management

Sirui Qi
Colorado State University
alex.qi@colostate.edu

Dejan Milojicic
Hewlett Packard Labs
dejan.milojicic@hpe.com

Cullen Bash
Hewlett Packard Labs
cullen.bash@hpe.com

Sudeep Pasricha*
Colorado State University
sudeep@colostate.edu

ABSTRACT

Today's cloud data centers are often distributed geographically to provide robust data services. But these geo-distributed data centers (GDDCs) have a significant associated environmental impact due to their increasing carbon emissions and water usage, which needs to be curtailed. Moreover, the energy costs of operating these data centers continue to rise. This paper proposes a novel framework to co-optimize carbon emissions, water footprint, and energy costs of GDDCs, using a hybrid workload management framework called SHIELD that integrates machine learning guided local search with a decomposition-based evolutionary algorithm. Our framework considers geographical factors and time-based differences in power generation/use, costs, and environmental impacts to intelligently manage workload distribution across GDDCs and data center operation. Experimental results show that SHIELD can realize 34.4× speedup and 2.1× improvement in Pareto Hypervolume while reducing the carbon footprint by up to 3.7×, water footprint by up to 1.8×, energy costs by up to 1.3×, and a cumulative improvement across all objectives (carbon, water, cost) of up to 4.8× compared to the state-of-the-art.

CCS CONCEPTS

• **Computer systems organization**; • **Architectures**; • **Distributed architectures**; • **cloud computing**;

KEYWORDS

Geo-distributed data centers, carbon emissions, wastewater, machine learning, evolutionary algorithms

ACM Reference Format:
Sirui Qi, Dejan Milojicic, Cullen Bash, and Sudeep Pasricha. 2023. SHIELD: Sustainable Hybrid Evolutionary Learning Framework for Carbon, Wastewater, and Energy-Aware Data Center Management. In *THE 14th international Green and Sustainable Computing Conference (IGSC '23), October 28–29, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3634769.3634810

*Corresponding author.

IGSC '23, October 28–29, 2023, Toronto, ON, Canada

ACM ISBN 979-8-4007-1669-0/23/10
https://doi.org/10.1145/3634769.3634810

1 INTRODUCTION

In recent years, the emerging use of general-purpose chat-bots, recommendation engines, and Internet-of-Things (IoT) devices [1] has increased the reliance on cloud data centers. Cloud service providers have been gradually distributing their data centers geographically across multiple locations. Such geo-distributed data centers (GDDCs) have many advantages. Establishing data centers closer to large customer bases offers better performance and lower network costs for them [2]. Multiple data centers also provide better resilience to catastrophic failures (e.g., environmental hazards).

However, thriving GDDCs are exacerbating the energy consumption and environmental impacts of cloud computing all over the world. Today, data centers account for 1% of worldwide electricity usage [3] and 0.6% of global greenhouse gas emissions [4]. There are over 2600 data centers in the United States and nearly 8000 data centers across the world, and this number is projected to increase in the coming decade [5]. These data centers consume large quantities of water, e.g., Google's 14 data centers consumed 4.3 billion gallons of water in 2021 [6], which puts immense pressure on local water supplies. Due to global climate change and the tightened energy policies in many nations [7], [8], researchers have recognized the need for realizing sustainable data centers.

Reducing the energy costs and environmental (water, carbon) overheads of GDDCs has thus taken on great urgency. From the perspective of minimizing energy costs, workloads should be assigned to data centers where there is cheap energy. Meanwhile, from the perspective of improving sustainability, workloads should be assigned to data centers that can provide cleaner (e.g., solar, wind) energy sources. These two perspectives are usually in conflict, and it is the cloud service provider's responsibility to manage workloads judiciously, so that both energy cost and sustainability goals can be met at the same time.

GDDCs provide compelling opportunities to better manage energy costs and environmental impacts [9]. For example, exploiting time-of-use (TOU) electricity pricing [10] can allow workloads to be executed at GDDC locations with lower TOU pricing (e.g., during off-peak periods), to reduce energy costs. Another opportunity is to utilize green energy techniques such as free air cooling [11] which may be available at some locations with environmental conditions

according to ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) [12]. Thus, to optimize GDDC operation, an effective GDDC management policy must consider time- and geography-based differences across data centers.

Prior efforts on GDDC management problem have focused on either minimizing energy costs (e.g., [13]) or an isolated sustainable goal such as carbon minimization or water-use minimization (e.g., [14], [15]). However, from a cloud service provider's perspective, there is a need to simultaneously optimize for all of these goals.

To address these important challenges, in this work, we propose a novel and efficient multi-objective optimization framework to co-optimize carbon emissions, water footprint, and energy costs of GDDCs. Our proposed Sustainable Hybrid Evolutionary Learning Framework for Geo-Distributed Data Center Management (SHIELD) performs intelligent design space exploration to generate efficient Pareto-optimal solutions that minimize the energy costs and environmental impact of GDDCs. The novel contributions of our work can be summarized as follows:

- We comprehensively model the carbon emissions, water profile, and energy use of GDDCs.
- We formulate a three-objective optimization problem for sustainable GDDC operation which involves minimizing carbon emissions, water footprint, and energy costs.
- We propose a new framework called SHIELD that combines machine learning and evolutionary algorithmic techniques to co-optimize the three objectives for GDDCs.
- We compare SHIELD with the state-of-the-art data center management frameworks and show that SHIELD outperforms them in speed and solution quality.

2 RELATED WORK

Cloud resource management has been studied for many years [16]. Single-objective challenges in cloud management such as quality of service [17], cost [18], [19], [20], resource utilization rate [21], performance [22], and fault rate [23] have been addressed by different methods. Several multi-objective optimization techniques have also been applied to cloud management in recent years, including simulated annealing (SA), genetic algorithm (GA), and non-dominated sorting [13], [24], [25].

Liu et al. proposed a holistic optimization framework for mobile cloud workload computing [24], aiming at triple-objective optimization (TOO) of energy consumption, quality of service, and system reliability. Their framework used SA, which was shown to be effective in providing trade-offs across the three objectives.

Hogade et al. proposed the genetic algorithm load distribution (GALD) approach to optimize energy costs in GDDCs [13]. This framework explored the potential of GDDCs, such as TOU electricity price, peak shaving, net metering, and local renewable energy availability. Moreover, GALD can be extended to a multi-objective problem [26] but can have slow convergence rates [27].

Bi et al. proposed a decomposition-based multi-objective algorithm with Gaussian mutation and crowding distance (DMGC), which co-optimized cost and revenue of workload scheduling in data centers [25]. Compared with GA, they were able to preserve more diverse designs in their solution set. The diversity in the population further benefited subsequent mutations and led to better final solutions. Furthermore, Gaussian mutation helped DMGC to jump out of local optima and converge to better solutions. However, our analysis indicates that the link between crowding distance and solution quality is weak. Crowding distance can filter out some high-quality designs and may select designs in unneeded directions.

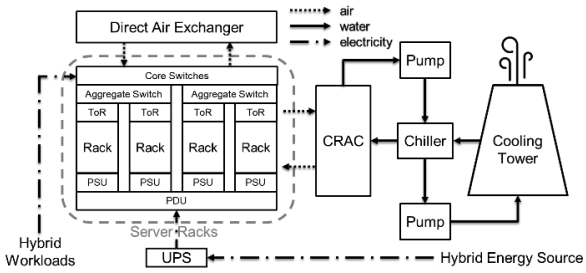

Figure 1: Air/Water/Electricity flow in a data center. Top-of-rack (ToR) switch, power supply unit (PSU), power distribution unit (PDU), and uninterrupted power supply (UPS) form the power switching system.

Besides optimization, modeling performance factors and key constraints are vital for datacenter management such as execution deadline [28], thermal constraint [29], [30], co-location impacts [31], [32]. Our proposed framework (SHIELD) combines machine learning and evolutionary algorithms to overcome drawbacks of state-of-the-art frameworks for data center management. Further, we tackle a more complex multi-objective optimization problem and use a more comprehensive system model than prior works.

3 SYSTEM MODEL

Our framework involves the mapping of each workload from its origin to different geographic locations and then to different nodes inside a data center. We further characterize energy costs and the environmental impact triggered by each mapping. Meanwhile, our framework can co-optimize the energy cost and environmental impacts by adjusting mappings and corresponding energy payment plans. Fig. 1 illustrates a data center's air/water/electricity flow that is modeled in our framework. Each data center is comprehensively modeled in terms of its power use, carbon emissions, water use, energy costs, and workload, as discussed in the rest of this section.

3.1 Power Model

3.1.1 Datacenter Layout. Each data center consists of N computing nodes, arranged in rows of racks [33]. The racks are arranged in a standard hot-aisle/cold-aisle configuration [34]. We assume several computing node types in data centers, whose number varies across locations. These nodes have different energy profiles for different computing workloads. Such heterogeneity in the GDDC node configurations is becoming widespread due to diverse workloads and service level agreements. Computer room air conditioning (CRAC) units are used to cool the data center room. Besides CRAC, we consider free air cooling availability in a subset of data centers. Free air cooling pumps cold outdoor air directly into the data center through direct air exchangers. We assume that these direct air exchangers are equipped with high-performance filters [35] to

resist air pollution. When the outdoor temperature and dew point both allow, the data center will switch to free air cooling from mechanical cooling [11].

3.1.2 Power Components. We divide power consumption of a data center i into information technology load $P_{IT,i}$, cooling $P_{Cooling,i}$, and Internal Power Conditioning System (IPCS) $P_{IPCS,i}$ [36].

IT load tracks the power consumption of servers in the data center and is related to the executed workload. Consider t different workload types computed by a active nodes. We assume a fixed p-state for each node. The power consumption of the IT load can be calculated by Eq. (1) below, where AP represents active power of workload type i assigned at node j, and IP is idle power of node k:

$$P_{IT} = \sum_{i}^{a} \sum_{j}^{t} AP_{ij} + \sum_{k}^{N-a} IP_k \tag{1}$$

There are three components of cooling power: CRAC units, chillers, and supporting equipment such as cooling tower, pumps, etc. [37]. From [33], the coefficient of performance (CoP) is the ratio of removed heat to the amount of necessary power to remove the heat. The CRAC power consumption can be estimated as:

$$P_{CRAC} = P_{IT}/CoP \tag{2}$$

From [37], the power consumption of chillers and the supporting equipment is close to the power consumption of CRAC units. Hence, we can estimate the whole cooling power as:

$$P_{Cooling} = 3 \times P_{CRAC} = 3 \times P_{IT}/CoP \tag{3}$$

The IPCS comprises power management components such as PDU, PSU, and UPS. The power consumption of IPCS correlates to the IT load and can be estimated as [36]:

$$P_{IPCS} = 0.13 \times P_{IT} \tag{4}$$

3.2 Water Model

We consider both site-based (from cooling) and source-based (from electricity generation) water consumption for data centers. By doing so, we can evaluate how geography-based differences impact water consumption. The overall water footprint of D GDDCs can be calculated by Eq. (5) below, in which $V_{E,i}$, $V_{B,i}$, and $V_{S,i}$ are volumes of evaporative, blowdown-to-wastewater-facility, and source water consumption of data center i respectively:

$$V_{ALL} = \sum_{i}^{D} \left(V_{E,i} + V_{B,i} + V_{S,i}\right) \tag{5}$$

Direct water consumption is common in mechanical cooling data centers, where the water is used as a coolant. Incoming water to data centers is usually potable water from water plants, while outgoing water from data centers is considered industrial wastewater, which needs to be processed at a wastewater facility [38]. We estimate the volume of direct water consumption through all means of water outflows, which are evaporative water through the cooling tower and blowdown water to the wastewater facility. The evaporative water consumption V_E through a cooling tower can be calculated by Eq. (6) below. E_{IT} represents the heat generated by IT infrastructure and H_{water} is latent heat of the water.

$$V_E = E_{IT}/H_{water} \tag{6}$$

The second part of site-based water outflow is the volume of blowdown water $V_B$to the wastewater treatment facilities. We assume all data centers cycle potable water until the concentration of dissolved solids is roughly C times the supplied water [39]. Hence, we estimate the volume of blowdown water by Eq. (7):

$$V_B = V_E/(C-1) \tag{7}$$

Source-based water consumption V_S is primarily from electricity generation which utilizes brown energy sources. Modern power grids usually utilize different energy sources and their brown energy ratios vary across locations. For example, the energy water intensity factor ($EWIF$) in Maryland is 0 while in Illinois it is 3.97 L/kWh [40], i.e., 3.97 liters of water are used when generating 1 unit of electricity in Illinois. We can calculate source-based water consumption based on energy Econsumed at data center as:

$$V_S = E \times EWIF \tag{8}$$

3.3 Carbon Model

Carbon dioxide is one of the biggest sources of greenhouse gas emissions [41]. Prior efforts on data center carbon emission reduction solely consider the minimization of electricity-based carbon emissions and ignore water-use-based carbon emissions. In this work, one of our novel contributions is to find correlations over carbon, water, and energy use in data centers, and co-optimize these. We thus analyze the carbon footprint from not just electricity generation but also potable water usage and wastewater treatment. Our analysis reveals that data centers with mechanical cooling may have a larger carbon footprint than expected. The overall carbon emission of D GDDCs can be calculated by Eq. 9) below, where $M_{electricity,i}$and $M_{water,i}$ are the mass of electricity-based and water-based carbon emitted at the data center i respectively:

$$M_{ALL} = \sum_{i}^{D} \left(M_{electricity,i} + M_{water,i}\right) \tag{9}$$

As geographical differences introduce energy source differences when calculating the electricity-based carbon emission, Carbon Factor CF measures the mass of emitted carbon during the process of electricity generation. We use the geographical CF from [42], and formulate the estimation function in Eq. (10), in which $M_{electricity}$ is the mass of electricity-based carbon emissions and E_B is the amount of brown energy used in a data center:

$$M_{electricity} = E_B/CF \tag{10}$$

We also characterize water-based carbon emissions due to the production of potable water and the treatment of wastewater. The water plant and wastewater treatment facility are assumed to use electricity from the local power grid. By combining geographical $EWIF$ [43] and CF, we obtain the water-based carbon emission from Eq. (11) below where I_P and I_W are the energy intensities for potable water production and wastewater treatment (representing the energy consumption per unit of water treatment):

$$M_{water} = \left[\left(V_B + V_E\right) \times I_P + V_S \times I_W\right]/CF \tag{11}$$

3.4 Energy Cost Model

We consider three price models that are relevant to the energy costs associated with distributing workloads across GDDCs: (a) TOU price, (b) clean premium, and (c) annual clean contract.

As discussed earlier, TOU price is a key factor that can help determine when to shift workloads to off-peak periods at different time zones, to reduce the energy cost of computing. However, this can lead to high environmental impacts because cheap power may not always be green power. For example, off-peak periods occur usually at midnight when there is no solar energy available.

Clean premium is an extra fee that cloud service providers pay for clean energy from the power grid. Once the extra fee is paid over the original TOU electricity price, the power provider can provide electricity from renewable energy sources. The premium price model is already available in many local power markers such as San Francisco and Denver where it allows cloud service providers to balance energy costs and environmental impacts.

Another cheaper clean energy price model is the annual clean contract, which exists in the Texas area in the United States. Several power providers such as Gexa Energy in Texas provide 24-hour all-year-around electricity from renewable sources. Compared with a clean premium, an annual clean contract can be cheaper if the annual energy can be estimated in advance.

3.5 Workload Model

We consider a rate-based workload management scheme [44], where the workload arrival rate can be estimated over a decision interval called an epoch [45]. In our work, epoch length is one hour, and thus a 24-epoch period represents a full day. Within the short duration of each epoch, workload arrival rates can be reasonably approximated as constant [46]. As shown in Eq. (12), our GDDC management framework needs to map the global arrival rate GAR_i of workload j into local arrival rates $AR_{i,d}$ across the D GDDCs:

$$GAR_j = \sum_{i}^{D} AR_{i,j} \tag{12}$$

4 PROBLEM FORMULATION

We consider a cloud service provider managing GDDCs across multiple locations inside USA. A GDDC management framework must distribute the workload coming in from various locations to data centers in the cloud service providers' GDDC. In each epoch (hour), the framework is responsible for providing distribution plans that include two parts: *(i)* workloads assigned to each location, and *(ii)* the amount of clean energy used at each location. The goal of the framework is to co-optimize three objectives: energy cost, carbon emissions, and water footprint. In our initial assumptions, the GDDCs are under-subscribed in the sense that they are expected to have enough computation resources to prevent any workload from being dropped or terminated before completion. The workloads originate off-site from the data centers, and we consider only workloads with negligible transfer time and costs. Once workloads are assigned to a data center, the same local data center scheduling policy is used to schedule local workloads no matter the location. The local scheduling policy is primarily based on workload type and node type. It builds on the list scheduling approach, where an ordered list of available heterogeneous nodes based on execution times is maintained per workload type, to guide the mapping [47].

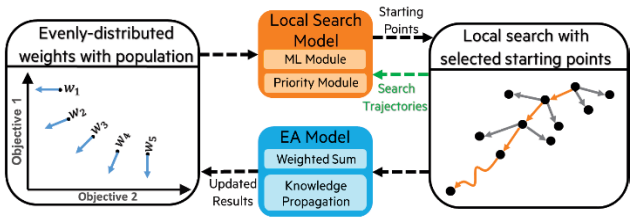

Figure 2: SHIELD framework overview

5 SHIELD FRAMEWORK

Our proposed SHIELD framework integrates a novel hybrid search approach that utilizes Machine Learning (ML)-guided local search with priorities and a Decomposition-based Evolutionary Algorithm (EA) with Knowledge Propagation. As shown in Fig. 2, a randomly generated population is input to an ML module for local search starting point selection. Due to a lack of training data in early iterations, the ML module randomly picks starting points in the beginning. After some time, these points are locally searched by the local search model based on improvement in their weighted sums. After local search, the population is updated with local search results and their trajectories are stored for ML module training. Our EA model further explores the design space and helps the local search model jump out of its local optima. After our EA updates the population, a new iteration starts with the ML module selecting the starting points. Algorithm 1 summarizes the pseudo-code of ML-guided local search (lines 2-7) and decomposition-based EA with knowledge propagation (lines 8-9). The input includes maximal generations for optimization gen, population size N, number of objectives M, number of early iterations for random local search $iter_{early}$, and ML module update frequency f_{update}. The output consists of N Pareto optimal designs for M objectives. The objective values of each design p are calculated based on models in Section 3.

5.1 Decomposition-based EA with KP

Unlike decomposition-based EA such as [48] which performs crossover and mutation only in a neighboring local space, our EA model realizes knowledge propagation (KP) by performing crossover across locally-searched points P_{start} and non-locally-searched points P_{rest} (line 8). Subsequently, our model mutates the crossover offspring to further explore the design space. Thus, our EA model guides crossover and mutation between locally-searched points and non-locally-searched ones, with knowledge gained during local search being propagated to non-locally-searched points, which avoids performing a local search on the whole population while still improving the overall population quality by expanding the exploration space. The generated offspring are used to update the population (line 9) via the function $Update(P, p, W)$, where each design point p in offspring $P_{offspring}$ is randomly compared with design points in P with the weighted sum function:

$$minimize\ g(x|w,z) = \Sigma_{i=1}^{M} \{w_i |Obj_i(x) - z_i|\} \tag{13}$$

Algorithm 1 SHIELD framework

Input: $gen,\ N, M,\ iter_{early},\ f_{update}$
Output: Population P (Final N designs)
Initialization:
Evenly Distributed Weight Vector Set $W = \{w_1, \ldots, w_N\}$
Randomly Generated Population $P = \{p_1, \ldots, p_N\}$
Training Set for ML Module $S_{train} \leftarrow \emptyset$
Ideal Point in M-objective Design Space $z = [o_1, \ldots, o_M]$
1: **for** $i = 0$ to gen **do**
2: **if** $i < iter_{early}$: $P_{start} \leftarrow Random(n_{local}, P)$
3: **else if** $(i - iter_{early})\ \%\ f_{update} == 0$:
$Eval \leftarrow MLtrain(S_{train})$; $S_{train} \leftarrow \emptyset$
4: **else**
$P_{start} \leftarrow MLguide(eval,\ W,\ P)$; $P_{start} = P_{start} \cup P_{priority}$
5: $P_{new} \leftarrow \emptyset$
6: **for** p, w in P_{start} **do**
$p_{new}, S_{trsj} = LocalSearch(w, p)$; $P_{new} = P_{new} \cup \{p_{new}\}$
$S_{train} = S_{train} \cup S_{traj}$
end for
7: $P = (P - P_{start}) \cup P_{new}$
8: $P_{rest} = P - P_{start}$; $P_{offspring} = EA(P_{start}, P_{rest})$
9: **for** p in $P_{offspring}$ **do** $P \leftarrow Update(P, p,\ W)$ **end for**
10: **end for**; **return** P

Unlike [49] which uses the Tchebycheff approach, this weighted sum approach uses a set of N uniformly spread weight vectors $W = \{w_1, \ldots, w_N\}$ in the following manner:

$$minimize\ g(x|w, z) = \max_{1 \le i \le M} \{w_i\ |Obj_i(x) - z_i|\} \qquad (14)$$

where g is the scalar optimization problem, M is the number of objectives, $Obj_i(x)$ is the i^{th} objective value of input x, and $Z = \{z_1, \ldots, z_m\}$ is the ideal point defined as the minimum value of all the objectives. Given a weight vector w_i, a lower Tchebycheff value $g(x|w, z)$ means a better design is found for the i^{th} subproblem. Instead of this approach, the weighted sum approach from Eq. (13) is deployed in our update function (line 9), to help our EA explore the design space. Compared with [49], our weighted sum approach can provide more diverse and fine-grained optimization directions (see results in Section 6) during design space traversal.

5.2 ML-guided Local Search with Priority

To boost the EA model both in speed and quality, a local search model is introduced in our framework. We observed that a local search model not only speeds up convergence, but also potentially provides much better individuals for EA to select as parents and then generate better offspring. Meanwhile, inspired by STAGE [50] which selects local search starting points using an evaluation function, an ML module is integrated into this framework to predict local search results (weighted sum) by studying previous local search trajectories (visited designs and weight vectors).

The local search model starts with a random local search (line 2) to create a training dataset S_{train} for subsequent ML module training $MLtrain(S_{train})$. The dataset contains search trajectories and is recreated at the frequency f_{update} after being used for ML module training (line 3). By using f_{update} , not only can we reduce the time redundancy introduced by ML module training, but we also keep S_{train} updated and compact. The ML module we use is a random forest model, which is an ensemble model that uses the average output from a collection of decision trees to help reduce overfitting. The evaluation function *eval* of the module maps each design's parameters and weight to the result of the search (Eq. (13)).

After $iter_{early}$ iterations, the local search model performs *eval* on all design points in P and selects local search starting points P_{start} based on predicted weighted sum improvement over the current weighted sums of P. This starting point selection process is represented by $MLguide(eval, W, P)$ in line 4. In this manner, we select local search starting points with the most potential for improvement. Meanwhile, a set of priority design points $P_{priority}$is added to P_{start} even though they have a smaller predicted weighted sum improvement than the original P_{start}. $P_{priority}$ is an enhancement to force local search in desired directions. These directions are usually single objective-efficient and have less weighted sum improvement in local searches. However, we find that searches on single objective-efficient directions can better explore the design space and improve the quality of non-locally-searched points P_{rest} through the EA model.

At each generation, an empty set P_{new} is used to store all endpoints p_{new} from local search (line 5). Each point p in P_{start} is then input to the local search function $LocalSearch(\mathrm{w}, p)$ and the function returns search endpoint p_{new} and search trajectory S_{traj} which is recorded in the training set S_{train} that is aggregated over generations (line 6). Lastly, all local search endpoints replace their corresponding starting points in P (line 7) to create an enhanced population that improves the outcomes from our EA approach.

6 EXPERIMENTS

6.1 Experiment Setup

We compare our proposed SHIELD with three state-of-the-art approaches: simulated annealing-based tri-objective optimization (TOO) [24], genetic algorithm-based load distribution (GALD) [13], and decomposition-based multi-objective evolutionary algorithm with Gaussian mutation and crowding distance (DMGC) [25]. All data centers use three different types of Intel server nodes: E3-1225v3, E5649, and E5-2697v2. These three nodes differ in their number of cores, frequency, power profile, and memory. These three types of nodes constitute 4320 computing nodes in each data center and their mix differs across locations. Due to the popularity of data analytics workloads among cloud service providers, we use 5 data-intensive workloads from the BigDataBench 5.0 [51], which are LDA, K-means, Naïve Bayes, image-to-text, and image-to-image workloads. In the power model, we configure CoP to be $3.75 \sim 5.72$ [33]. In the water model, water latent heat and $EWIF$ are set as $0.66\ kWh/L$ (at40°C) and $0 \sim 3.97\ L/kWh$ [40]. In the carbon model, I_P and I_W are $550 kWh/ML$ and $640 kWh/ML$ [43]. The other parameters such as concentration cycle (C), CF, TOU, clean premium, and annual clean contract are configured to be 5, $99.7 \sim 775 g/kWh$ [42], $1.8 \sim 48¢/kWh$, $0.39 \sim 144¢/kWh$, and $15¢/kWh$ respectively. We consider 16 different data center locations, with diverse characteristics, as shown in Fig. 3.

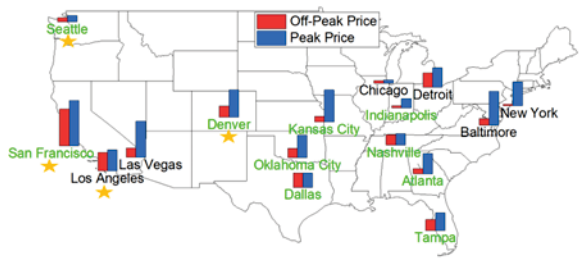

Figure 3: Data center power price map with different price models: TOU price, clean premium, and annual clean contract. The red/blue bars indicate the local off-peak/peak power price, which is 1.8~48 ¢/kWh. Locations in green are ones with clean premium projects available. The starred locations are equipped with air-free cooling techniques which switch to air-free cooling in lower-than-75°F temperatures and lower-than-63°F dew points.

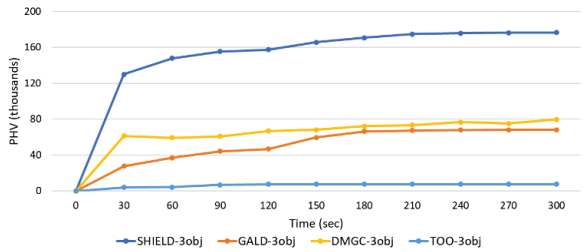

Figure 4: Three objective (energy cost, carbon, water) PHV over time of SHIELD, GALD, DMGC, and TOO.

6.2 Experiment Results

6.2.1 PHV Improvement. To compare the quality of solutions generated by each framework, we first determined the PHV of generated solutions over time for a scenario with 16 data centers and all three objectives (minimizing energy cost, carbon footprint, and water footprint). PHV measures the size of the space enclosed by all solutions on the Pareto front and a user-defined reference point. A higher value of PHV is indicative of a more diverse and higher-quality solution set.

From Fig. 4, we can observe that all frameworks converge within ~3 minutes. The PHV of SHIELD is 2.1× larger than that of the second-best framework (GALD). Further, to reach the second-best PHV result, it takes SHIELD 34.4× less time than GALD. Based on these results, we can see that SHIELD optimizes the PHV better and faster than other design space exploration frameworks.

6.2.2 Solution Quality Improvement. We analyzed solution quality across our three metrics for a scenario with 16 data centers and a 1-minute runtime constraint, to ensure real-time decision making at the beginning of each epoch.

Fig. 5 shows results aggregated over a 24-hour interval, with the y-axis showing improvements compared to the TOO framework. The three best solutions are selected from each framework. The most energy cost-efficient solution is depicted with the first three

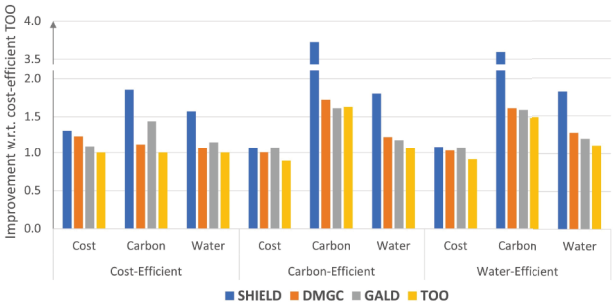

Figure 5: 24-hour aggregated results for cost-, carbon-, and water-efficient solutions generated across four frameworks.

sets of bars that show the energy cost, carbon emissions, and water use of the most energy cost-efficient solution selected from each of the four compared frameworks. Similarly, the three middle sets of bars characterize the most carbon-efficient solution, while the last three sets of bars characterize the most water-efficient solution.

From Fig. 5, it can be observed that SHIELD is always the best in cost/carbon/water reduction no matter which of the three metrics is prioritized. For the most energy cost-efficient solution (first three sets of bars), SHIELD has higher cost reduction as well as lower carbon emission and water consumption compared to all other frameworks. Similarly, for the most carbon-efficient solution (middle three sets of bars) and the most water-efficient solution (last three sets of bars), SHIELD generated higher quality solutions that outperform those from other frameworks. SHIELD reduces energy costs, carbon emissions, and water usage by up to 1.3×, 3.7×, and 1.8× respectively. We also determine a single best solution with the lowest cumulative energy cost, carbon emission, and water usage for each epoch, for each framework. Over a 24-hour interval, we found that SHIELD improves cumulative solution quality by 4.8×, 2.4×, and 3.2× when compared to TOO, GALD, and DMGC.

7 CONCLUSION

In this work, we studied the problem of workload distribution across geo-distributed data centers (GDDCs) to minimize energy cost, carbon footprint, and water use, simultaneously. We developed comprehensive models of energy consumption, energy price, water consumption, carbon emission, and workload execution. We then developed a novel framework called SHIELD for multi-objective optimization of the cloud service providers' workload distribution problem.. In our experiments, SHIELD was able to realize 34.4× speedup and 2.1× improvement in PHV while reducing the carbon footprint by up to 3.7×, water footprint by up to 1.8×, energy costs by up to 1.3×. and a cumulative improvement across all objectives (carbon, water, cost) of up to 4.8×, compared to state-of-the-art frameworks.

ACKNOWLEDGMENTS

This research was made possible with support from HPE and grants from the National Science Foundation (CCF-2324514, CNS-2132385).

REFERENCES

[1] Amazon, "Global Infrastructure," [Accessed 1 May 2023], Available: http://aws.amazon.com/about-aws/global-infrastructure/.

[2] N. Hogade and S. Pasricha, "A Survey on Machine Learning for Geo-Distributed Cloud Data Center Management," in IEEE TSUSC, 2023.

[3] E. Masanet, *et al.*, "Recalibrating global data center energy-use estimates," Science, vol. 367, no. 6481, pp. 984-986, 2020.

[4] IEA, "Data Centres and Data Transmission Networks," [Accessed 1 May 2023], Available: https://www.iea.org/reports/data-centres-and-data-transmission-networks.

[5] B. Daigle, "Data Centers Around the World: A Quick Look," United States International Trade Commission, 2021.

[6] Google, "Google Data Centers 2021 Annual Water Metrics," 2021.

[7] T. E. Wirth, *et al.*, "The future of energy policy," Foreign Affairs, 2003.

[8] F. Libertson, *et al.*, "Data-Center infrastructure and energy gentrification: perspectives from sweden," SSPP, vol. 17, no. 1, 2021.

[9] S. Pasricha, N. Hogade, H.J. Siegel, and A. A. Maciejewski, "Green Computing with Geo-Distributed Heterogeneous Data Centers," in IGSC, 2022.

[10] Y. Li, H. Wang, *et al.*, "Operating cost reduction for distributed internet data centers," IEEE/ACM CCGRID, 2013.

[11] J. Ni, *et al.*, "A review of air conditioning energy performance in data centers," Renewable And Sustainable Energy Reviews, 2017.

[12] ASHRAE, "Equipment thermal guidelines for data processing environments," 2021.

[13] N. Hogade, S. Pasricha, H. J. Siegel, A. A Maciejewski, M. Oxley, and E. Jonardi, "Minimizing energy costs for geographically distributed heterogeneous data centers," IEEE TSUSC, 2018.

[14] T. A. Ndukaife, *et al.*, "Optimization of water consumption in hybrid evaporative cooling air conditioning systems for data center cooling applications," Heat Transfer Engineering, vol. 40, no. 7, pp. 559-573, 2019.

[15] S. Ruth, "Reducing ICT-related carbon emissions: an exemplar for global energy policy," IETE technical review, vol. 28, no. 3, 2021.

[16] B. K. Dewangan, *et al.*, "Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges," Software: Practice and Experience, vol. 51, 2021.

[17] S. K. Garg, *et al.*, "SLA-based virtual machine management for heterogeneous workloads in a cloud," J. Netw. Comput. Appl., vol. 45, 2014.

[18] H. Yuan, *et al.*, "Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds," IEEE T-ASE, vol. 14, no. 1, 2017.

[19] N. Hogade, S. Pasricha, and H.J. Siegel, "Energy and Network Aware Workload Management for Geographically Distributed Data Centers," IEEE Transactions on Sustainable Computing (TSUSC), vol. 7, no. 2, 2022.

[20] E. Jonardi, M. Oxley, S. Pasricha, H. J. Siegel and T. Maciejewski, "Energy Cost Optimization for Geographically Distributed Heterogeneous Data Centers," in 2015, IEEE Workshop on Energy-efficient Networks of Computers (E2NC): from the Chip to the Cloud.

[21] D. Alsadie, *et al.*, "Dynamic resource allocation for an energy efficient VM architecture for cloud computing.," ACM ACSW, vol. 16, 2018.

[22] D. Dauwe, S. Pasricha, A. A. Maciejewski, and H. J. Siegel, "HPC Node Performance and Energy Modeling Under the Uncertainty of Application Co-Location," Journal of Supercomputing, vol. 72, no. 12, 2016.

[23] D. Dauwe, S. Pasricha, A. A. Maciejewski, and H. J. Siegel, "A Performance and Energy Comparison of Fault Tolerence Techniques for Exascale Computing Systems," in 6th IEEE International Symposium on Cloud and Service Computing (SC-2), 2016.

[24] H. Liu, *et al.*, "A holistic optimization framework for mobile cloud task scheduling," IEEE TSUSC, 2017.

[25] J. Bi, *et al.*, "Green energy forecast-based bi-objective scheduling of tasks across distributed cloud," IEEE TSUSC, 2021.

[26] H. Tamaki, *et al.*, "Multi-objective optimization by genetic algorithms: A review," IEEE ICEC, 1996.

[27] K. Sindhya, *et al.*, "A hybrid framework for evolutionary multi-objective optimization," IEEE TEVC, vol. 17, no. 4, pp. 495-511, 2012.

[28] M. Oxley, S. Pasricha, H. J. Siegel, and A. A. Maciejewski, "Energy and Deadline Constrained Robust Stochastic Static Resource Allocation," in Workshop on Power and Energy Aspects of Computation (PEAC) held in conjunction with the 10th International Conference on Parallel Processing and Applied Mathematics (PPAM), 2013.

[29] A. M. Al-Qawasmeh, S. Pasricha, A. A. Maciejewski, and H. J. Siegel, "Thermal-Aware Performance Optimization in Power Constrained Heterogeneous Data Centers," in 21st International Heterogeneity in Computing Workshop (HCW), 2012.

[30] M. Oxley, S. Pasricha, A. A. Maciejewski, H. J. Siegel and P. Burns, "Online Resource Management in Thermal and Energy Constrained Heterogeneous High Performance Computing," in IEEE International Conference on Big Data Intelligence and Computing (DataCom), 2016.

[31] D. Dauwe, E. Jonardi, R. Friese, S. Pasricha, A. A. Maciejewski, D. Bader, and H. J. Siegel, "A Methodology for Co-Location Aware Application Performance Modeling in Multicore Computing," in 17th Workshop on Workshop on Advances in Parallel and Distributed Computational Models (APDCM), 2015.

[32] M. Oxley, E. Jonardi, S. Pasricha, A. A. Maciejewski, G. Koenig and H. J. Siegel, "Thermal, Power, and Co-location Aware Resource Allocation in Heterogeneous Computing Systems," in IEEE International Green Computing Conference (IGCC), 2014.

[33] J. D. Moore, *et al.*, "Making scheduling "cool": temperature-aware workload placement in data centers," USENIX ATC, 2005.

[34] R. F. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms," Uptime Institute, 2000.

[35] H. M. Daraghmeh, *et al.*, "A review of current status of free cooling in datacenters," Applied Thermal Engineering, vol. 114, 2017.

[36] K. M. U.Ahmed, *et al.*, "A review of data centers energy consumption and reliability modeling," IEEE Access, vol. 9, 2021.

[37] Q. Zhang, *et al.*, "A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization," Journal of Systems Architecture, vol. 119, 2021.

[38] D, Azevedo, *et al.*, "Water usage effectiveness (WUE): A green grid datacenter sustainability metric," The Green Grid, 2011.

[39] M.A. B. Siddik, *et al.*, "The environmental footprint of data centers in the United States," Environmental Research Letters, vol. 16, no. 6, 2021.

[40] P. Torcellini, *et al.*, "Consumptive water use for US power production," NREL, 2003.

[41] S., Mohammad, *et al*, "Carbon dioxide separation from flue gases: a technological review emphasizing reduction in greenhouse gas emissions," Sci. World J., 2014.

[42] EIA, [Accessed 1 May 2023], Available: https://www.eia.gov/tools/faqs/faq.php?id$=$74&t$=$11.

[43] P. A. Malinowski, *et al.*, "Energy-water nexus: Potential energy savings and implications for sustainable integrated water management in urban areas from rainwater harvesting and gray-water reuse," J. Water Resour. Plan. Manag, 2015.

[44] M. Oxley, E. Jonardi, S. Pasricha, H. J. Siegel, A. A. Maciejewski, P. J. Burns, and G. Koenig, "Rate-based Thermal, Power, and Co-location Aware Resource Management for Heterogeneous Data Centers," in JPDC, 2018.

[45] J. Kumar, *et al.*, "BiPhase adaptive learning-based neural network model for cloud datacenter workload forecastin," Soft Computing, 2020.

[46] D. G. Feitelson, at al., "Experience with using the Parallel Workloads Archive," JPDC, vol. 74, pp. 2967-2982, 2014.

[47] S. Ghanbari, *et al.*, "A priority based job scheduling algorithm in cloud computing," Procedia Engineering, vol. 50, 2012.

[48] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," IEEE TEVC, vol. 11, no. 6, 2007.

[49] J, Andrzej, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment," IEEE Trans. Evol. Comput., vol. 6, no. 4, pp. 402-412, 2002.

[50] J. A. Boya, *et al.*, "Learning evaluation functions to improve optimization by local search," J. Mach. Learn. Res., pp. 77-112, 2001.

[51] "BigDataBench 5.0 benchmark suite," [Accessed 1 May 2023], Available: http://www.benchcouncil.org/BigDataBench/.