HOLISTIC, REAL-TIME OPTIMIZATION OF THE OPERATIONS OF DISTRICT COOLING SYSTEMS VIA DEEP REINFORCEMENT LEARNING AND MIXED INTEGER LINEAR PROGRAMMING

Zhonglin Chiam[1,2], Arvind Easwaran [1,*], David Mouquet [3], Mohit Gupta [4]

1 School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

2 Veolia City Modeling Centre (VECMC), 23 Pandan Avenue, Singapore 609389, Singapore

3 Veolia Environment Recherche et Innovation (VERI), 291 Avenue Dreyfous Ducas, 78520, Limay, France

4 Department of Computer Science and Engineering, I.I.T. Delhi, Hauz Khas, New Delhi 110016, India

ABSTRACT

The holistic optimization of district cooling systems is a computationally intensive undertaking, owing to the sheer number of conflicting decision variables and non-convex nature of the problem. This is the primary reason which inhibits the real-time deployment of optimization algorithms for the operations of district cooling systems. To overcome this challenge, we adopt a model-based, decomposed approach involving the concurrent use of reinforcement learning and mixed integer linear program to holistically optimize the thermal and physical interactions while still capturing the tight coupling between the components of the system. Resolution speed and solution accuracy are paramount for a real-time optimization algorithm thus, the critical advantage of the proposed approach is two-fold – the mixed integer linear program drastically reduces the action space of the reinforcement learning problem, promoting accuracy and when trained, the agent neural network can then rapidly determine the optimal values of the remaining actions, improving resolution speed.

The current work makes the two ensuing vital contributions: (1) we introduced a decomposed optimization approach with resolution speeds which are compatible with real-time deployment, (2) through the application on a real test-case, we compare both the resolution time and solution quality against an approach used in our previous work, which deployed the genetic algorithm instead of a reinforcement learner. Results indicate that the impact on solution quality is below 7.52%, thereby, validating the feasibility of the proposed approach.

Keywords: reinforcement learning, district cooling, energy efficiency, cyber-physical systems, holistic optimization, mixed integer linear program.

NONMENCLATURE

Symbols	
Δ	difference
δ	neural network error
%	percentage
θ	error of the neural network
a	actor neural network
c	critic neural network
$\dot{E}$	electricity consumption (kWh)
$\dot{m}$	volume flowrate (m^3/h)
μ	mean
$\mathcal{N}$	normal distribution
σ	standard deviation
P	pump
T	temperature (K)
Q	cooling demand (kWh)
X	input data into the neural network
Superscripts	
in	flows into the unit
out	flow out of the unit
Subscripts	
c	customer
ch	chiller
cp	common pipe
$cond$	condenser side of the chiller
ct_unit	cooling tower unit

Selection and peer-review under responsibility of the scientific committee of the 11th Int. Conf. on Applied Energy (ICAE2019).

DCS	district cooling system
$dist\_nwk$	distribution network
$evap$	evaporator network
sel	selected pump
sys	entire network system
$i, j, 1, 2, 3 \ldots$	numerical labels

1. INTRODUCTION

Space cooling, especially in the tropics account for up to 40% of energy demand [1]. District cooling systems (DCS) represent a potentially more efficient means of fulfilling this demand [2]. Therefore, they are increasingly becoming commonplace in newer urban developments [3]. However, the actual performance is quite uncertain due to the inevitable deviation between design and operating conditions [4]. Oversizing of DCS, due to erring on the side of caution often results in the gross under-utilization of equipment such as chillers, impacting the efficiency of the overall system very negatively [5, 6].

Inefficiency is further exacerbated by control strategies which are either predefined or seek only to optimize the performance of chillers [7]. These measures disregard the cascading effect on the system, impeding the ability of the DCS to respond well under unfavorable load conditions. Thus, this paper focuses on introducing a real-time optimization approach which not only thoroughly explores the solution space mapped by tuneable variables in a DCS, i.e., optimizes the system holistically, but additionally resolves swiftly enough such that it is compatible with deployment in real-time. The approach discussed in this paper is an improvement to the hierarchical optimization framework which we previously introduced [8].

1.1 Prior work

Reinforcement learning (RL) is a generic machine learning framework which involves an agent taking actions in an environment and thereby earning a reward for it. The goal of the agent is to maximize the cumulative reward from the environment by iteratively improving the actions taken. The successful deployment of RL in playing games (Go [9], Atari [10]) and image recognition [11] marks the most significant progress in recent history. Despite the progress, it is essential to note that most of these work deal with discrete state and action spaces which is not directly compatible with those requiring continuous state and action spaces [12]. Naïve discretization of the state or action space leads to the curse of dimensionality makes solving intractable. Thus, a class of RL algorithms known as deterministic policy gradients (DPG) has been introduced [13].

As a technique for optimal control, RL is rapidly gaining traction in the closely related field of building energy control [14, 15]. Examples of such application include the optimization of energy performance or operating cost in heating ventilating and cooling (HVAC) systems, domestic hot water (DHW) and data center cooling, through the manipulation of temperature setpoints. Since these problems mainly involve continuous state and action spaces, deterministic methods have been employed [16, 15]. These implementations typically use end-to-end model-free approaches which only deal with the optimization of a small set of actions (decision variables). There is however no indication that the optimization of the action space will yield optimal performance of the overall system. Training stability and convergence are other major issues which plague the performance of RL in these domains; hence techniques such as recurrent neural networks, experience replays, pretraining the neural networks with copious buffers of offline traces and guidance through expert defined policies were some of the measures adopted mitigate these issues.

1.2 Objectives and contributions

Holistic optimization of DCS itself presents a challenging task as a myriad of decision variables must be simultaneously optimized while respecting the tight coupling between the components of the system and non-linearities in the governing equations.

First, we chose appropriate models for representing each component of the DCS and calibrated them using raw data. Subsequently, these models were abstracted so that the resulting optimization problem could be solved using the combination of RL and mixed integer linear program (MILP). The MILP drastically reduces the action space of the RL problem, promoting convergence when training the neural network.

For illustrative purposes, we applied this approach to a test-case based on an existing DCS in Europe. Results generated were compared against an existing framework involving the combination of the genetic algorithm (GA) and MILP [8]. For the validating cooling demand scenario defined, results indicate that there is only up to 7.52% difference in the objective function while realizing a vastly significant resolution speed for real-time deployment.

Our proposed real-time optimization approach, its subsequent application on a test-case and validation against our previous approach are the two vital contributions of this paper.

1.3 *Organization of the paper*

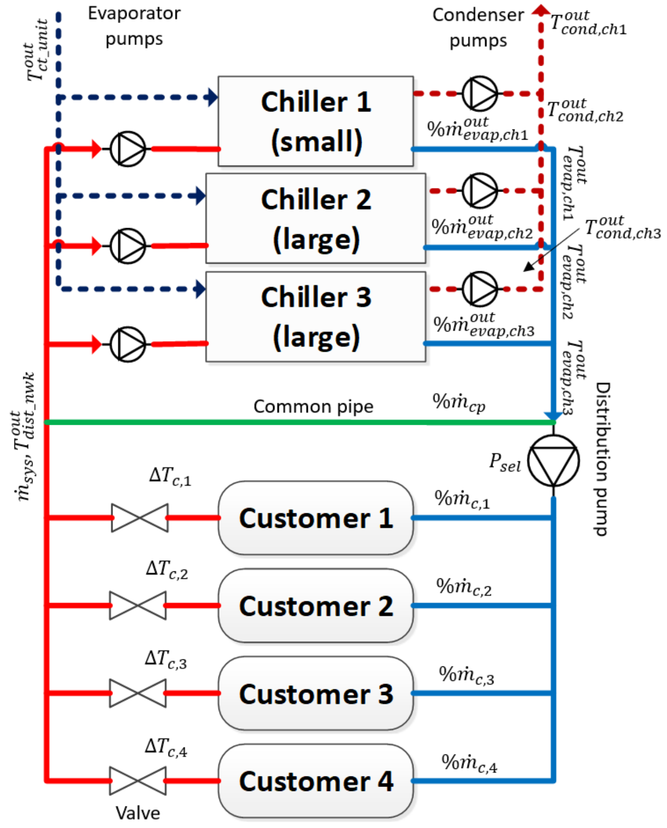

Figure 1: Schematic of DCS in the test-case

The next section introduces our RL-MILP approach by applying it on a test-case. Section *3* and *4* presents and analyzes the results of the test-case respectively, while Section *5* concludes our findings and discuss possible improvements for future work.

2. FORMULATION OF RL-MILP REAL-TIME OPTIMIZATION PROBLEM

The discussion in this section will take the following shape – first, we define the test-case, then discuss our previous work in brief, before concluding with our implementation of RL onto the test-case. The discussion of our previous work is essential as it forms the basis for our current work.

2.1 *Test-case description*

The test-case (*Figure 1*) presented in this sub-section is based on a functioning DCS located in Europe. The DCS of interest comprises of a single central station housing three water cooled chillers and cooling towers which serve four customers of different load profiles. As the DCS was overprovisioned, it is plagued by the infamous low *Δ*T syndrome, degrading its overall efficiency.

The objective of our work is to reduce the electricity consumption ($\dot{E}_{DCS}$) of the DCS by holistically optimizing operations at the hourly level. This involves determining the optimal values of all the variables listed in *Figure 1* for a given combination of cooling demand ($\dot{Q}_{d,c,i}$) and ambient temperatures (T_{wb}). We did not include cooling towers due to limitations in the dataset we had for calibrating the models.

Cooling demand for a single representative day will be used to both validate the performance of our current proposed against our previous approach. The quality of the optimal objective function and resolution speed will be the metric for measuring the performance of both methods.

2.2 *Previous work: GA-MILP approach*

In our previous work, after choosing the appropriate models, we decomposed the optimization problem into two-levels – master (GA) and a slave (MILP). Doing so, increased the likelihood of converging to the globally optimal solution as the prowess of MILP solvers can be leveraged. This approach is summarized in *Figure 2* and more details can be found in [8].

When we formulated the test-case, we found that only two decision variables ($\dot{m}_{sys}, T^{out}_{dist\_nwk}$) were required to be optimized by the GA, whilst the rest could be adequately handled by the MILP. The main drawback of the GA-MILP approach is the numerous iterations in which the GA must go through just to determine the optimal values of two decision variables. A typical run of the MILP takes under 5 seconds to solve, which is considerably fast, since we are solving for the optimal values to be used over a period of an hour. Coupling it

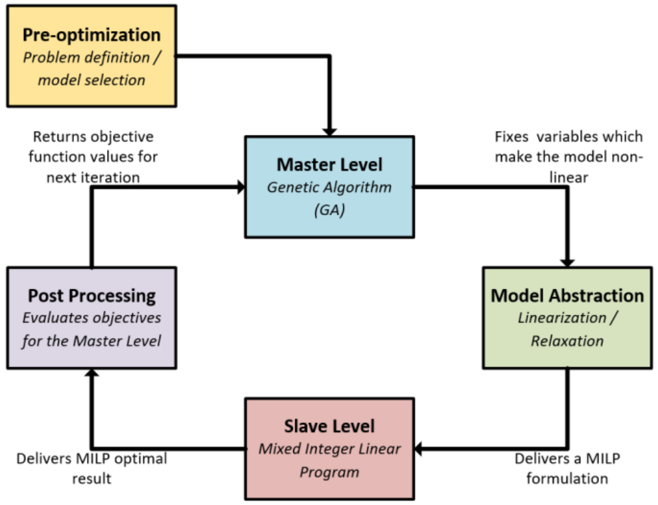
Pre-optimization
Problem definition / model selection
Returns objective function values for next iteration
Master Level
Genetic Algorithm (GA)
Fixes variables which make the model non-linear
Post Processing
Evaluates objectives for the Master Level
Model Abstraction
Linearization / Relaxation
Slave Level
Mixed Integer Linear Program
Delivers MILP optimal result
Delivers a MILP formulation

Figure 2: Summary of our previous work: GA-MILP approach

with the GA in the manner shown in *Figure 2* greatly retards the entire process. Should the relationship between $\dot{m}_{sys}$ and $T_{dist\_nwk}^{out}$ to the optimal electricity consumption be determined offline, we could utilize this as a real-time optimization framework to aid the decision making process for operators of the DCS. This underpins the motivation for our next approach.

2.3 RL-MILP approach

The first step toward utilizing RL is to formulate the problem as a Markov decision process (MDP), with the proper definition of the environment, reward, state and action spaces. With added assumption that the actions taken in each hourly time-step in the test-case is independent from the next, the optimization problem could be formulated as in the similar fashion as the classic multi-armed contextual bandit problem – the only difference being continuous state and action spaces [17]. Hence, we replaced the GA with our variant of the deep deterministic policy gradient (DDPG) algorithm [11], however, instead of a reward, we introduced a negative reward to discourage the agent from making poor choices. Without using the decomposition approach, the subsequent RL problem may be difficult to solve as the action space is too large for the state space. *Table 1* details our definition of the MDP.

State	$Q_{c,1-4},\ T_{wb}$
Action	$\dot{\mathrm{m}}_{\mathrm{sys}},\mathrm{T}_{\mathrm{dist\_nwk}}^{\mathrm{out}}$
Environment	*MILP Formulation*
Negative reward	*Objective function* $(\dot{E}_{DCS})$

Table 1: Definition of the MDP

We implemented the actor-critic method for our RL design. The actor intakes a state and outputs an action, and the critic evaluates the state and corresponding action from the actor. *Figure 3* illustrates the implementation of our forward feed neural network architecture and *Figure 4* summarizes our proposed RL-MILP approach. For the actor neural network, we defined the final hidden layer to output four values – two mean (μ) and two sigma (σ) values which was then used with the normal distribution function to generate the values for the output layer. After which, we reduced the losses of our neural networks using the Adam optimizer, with learning rates of 0.001 and 0.0001 for the actor and critic network respectively [18]. Finally, *Algorithm 1* details training procedure we used to update the weights for the neural networks.

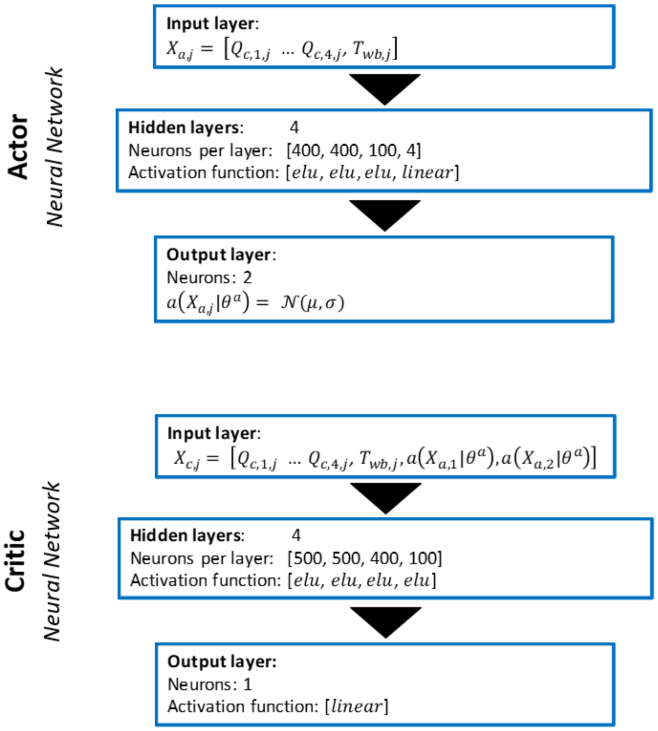
Actor
Neural Network
Input layer:
$X_{a,j} = [Q_{c,1,j} \dots Q_{c,4,j}, T_{wb,j}]$
Hidden layers: 4
Neurons per layer: [400, 400, 100, 4]
Activation function: $[elu, elu, elu, linear]$
Output layer:
Neurons: 2
$a(X_{a,j}|\theta^a) = \mathcal{N}(\mu,\sigma)$
Critic
Neural Network
Input layer:
$X_{c,j} = [Q_{c,1,j} \dots Q_{c,4,j}, T_{wb,j}, a(X_{a,1}|\theta^a), a(X_{a,2}|\theta^a)]$
Hidden layers: 4
Neurons per layer: [500, 500, 400, 100]
Activation function: $[elu, elu, elu, elu]$
Output layer:
Neurons: 1
Activation function: $[linear]$

Figure 3: Neural network architectures

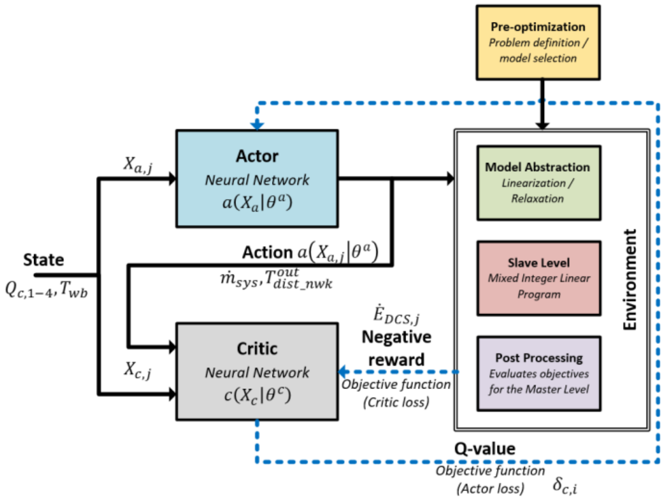

Figure 4: Summary the RL-MILP approach

3. RESULTS AND DISCUSSION

In this section, we compare the results of holistic optimization of DCS operations using the GA-MILP and RL-MILP approach from two perspectives – solution quality and resolution speed.

3.1 Solution quality

Figure 5 illustrates compares the $\dot{E}_{DCS}$ values generated for the validation data trace using the both approaches against the base-case. After training the RL for 3000 episodes, we can already notice significant savings in electricity consumption as compared to the base-case. Although the GA-MILP approach generally performed better, we note that there exist scenarios such as in the 1st, 5th and 17th hour where the RL-MILP approach marginally outperformed the GA-MILP. One possible explanation for this could be due to the gradient descent optimizers advantage over the GA when searching for local optima. The converse is true especially in the 12th and 22nd hour where it is likely that the optimizer used to minimize the losses in the RL is unable to escape a local optimum.

Algorithm 1: *Training algorithm for the actor-critic neural networks*

```
1. Collect the hourly state data over the period of 1
   month. The negative reward refers to $\dot{E}_{DCS}$ which
   is an output from the environment.
2. Select 1 day outside the training data trace to be
   used for validation.
3. Initialize neural networks $a(X_a|\theta^a)$ and
   $c(X_c|\theta^c)$ with weights $\theta^a$ and $\theta^c$ initialized
   using the Kaiming initialization scheme [19].
4. for $i = 1, \ldots, max\ episodes$ do
       Compute $\delta_{c,i} = \left(c\left(X_{c,i}|\theta^c\right) - \dot{E}_{DCS,i}\right)$
       Update $\theta^c$ by miimizing $\delta_{c,i}{}^2$
       Compute $\delta_{a,i} = -\log\left(\mathcal{N}\left(a\left(X_{a,i}|\theta^a\right)\right)\right) \times \delta_{c,i}$
       Update $\theta^a$ by minimizing $\delta_{a,i}$
   end for
```

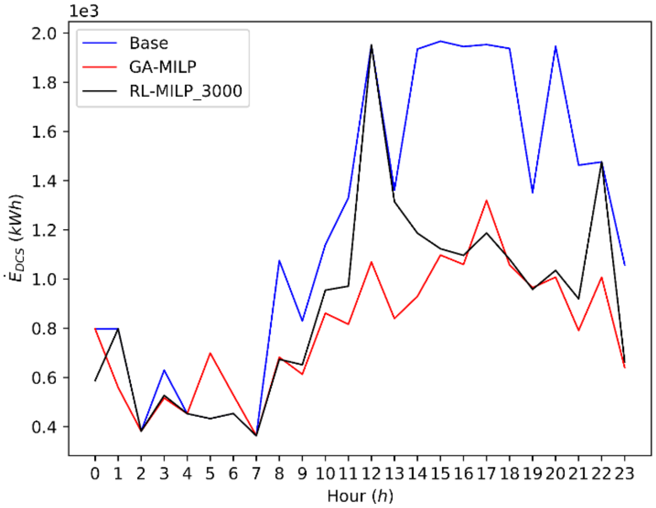

Figure 5: Plot of $\dot{E}_{DCS}$ (kWh) for the base-case, MILP-GA approach and RL-MILP (after 3000 episodes)

Table 2 documents the percentage difference of $\dot{E}_{DCS}$ of the RL-MILP against the base-case and the GA-MILP. The accuracy generally improves with the number of training episodes. Beyond 3000 episodes however, the performance on the validation data trace started to degrade. Overfitting to the training data trace could be a likely reason for this observation. Another possible explanation could pertain to the inability to fully capture the relationship between the states and actions within the current size and architecture of the neural networks. Where escaping local optima are concerned, off-policy methods could also be explored.

Episodes	***Percentage difference (RL-MILP vs Base-case)***	***Percentage difference (GA-MILP vs Base-case)***
1000	-3.92	-34.36
2000	-23.68	
3000	-26.84	
4000	-22.55	

Table 2: Comparison of difference in $\dot{E}_{DCS}$ (kWh) between the base-case and MILP-GA approach over

3.2 Resolution speed

This where the primary benefit of the RL-MILP approach lies. The average resolution time to solve the MILP sub-problem is approximately 3 - 5s. The GA requires about 15 000 evaluations of the MILPs to converge for a single time-step of an hour. Despite only having undergone 3000 MILP evaluations, the absolute difference in electricity savings between the GA-MILP and RL-MILP approaches is only 7.52%.

Since the RL is trained offline, online performance will not be impeded, regardless of the number of training episodes it requires to converge – when properly trained, a single evaluation of the MILP is all that is required to deliver the optimal values of all the decision variables required to operate the DCS efficiently.

4. CONCLUSION

We introduced an approach for the holistic optimization of the DCS operations using the combination of reinforcement learning and mixed integer linear program. Reinforcement learning can shift the majority of the heavy computation offline, vastly improving the feasibility of our proposed approach for real-time applications. When the reinforcement learner was trained, we could retrieve the close to optimum solutions almost instantly, a feat not achievable with our previous approach which utilized the combination of the genetic algorithm and mixed integer linear program. When the both methods were compared, only a mere

7.52% of electricity savings was forgone for an immense improvement in the resolution speed of the algorithm.

Accounting for stochasticity in cooling demand and the optimization capability of our reinforcement learner are listed as directions for our future work.

ACKNOWLEDGEMENT

The authors would like to acknowledge Veolia City Modeling Center (VECMC) for the financial support.

REFERENCE

[1] A. R. Katili, R. Boukhanouf and R. Wilson, "Space cooling in buildings in ot and humid climates - a review of the effects of humidity on the applicability of existing cooling techniques," in *14th International Conference on Sustainable Energy Technologies* , Nottingham, United Kingdom, 2015.

[2] Electrical and Mechanical Services Department, HKSARG, "Energyland," 07 25 2014. [Online]. Available: https://www.emsd.gov.hk/energyland/en/building/district_cooling_sys/dcs_benefits.html. [Accessed 12 05 2019].

[3] H. J. Uy, "District cooling is gaining momentum in Asia," Climate control middle east, Singapore, 2018.

[4] N. Deng, G. He, Y. Gao, B. Yang, J. Zhao, S. He and X. Tian, "Comparative analysis of optimal operation strategies for district heating and cooling system based on design and actual load," *Applied Energy,* vol. 205, pp. 577-588, 2017.

[5] M. Schwelder, "Using low-load chillers to improve system efficiency," *ASHRAE Journal,* pp. 14-20, 2017.

[6] C. Yan, W. Gang, X. Niu, X. Peng and S. Wang, "Quantitative evaluation of the impact of building load characteristics on energy performance of district cooling systems," *Applied Energy,* vol. 205, pp. 635-643, 2017.

[7] S. Wang, Intelligent buildings and building automation, United Kingdom: Taylor & Francis, 2009.

[8] Z. Chiam, A. Easwaran, D. Mouquet, S. Fazlollahi and J. V. Millás, "A hierarchichal framework for holistic optimization of the operations of district cooling systems," *Applied Energy,* vol. 239, pp. 23-40, 2019.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, pp. 529-533, 2015.

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature,* vol. 529, pp. 484-489, 2016.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning," in *International conference on learning representations*, San Juan, Puerto Rico, 2016.

[12] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature,* vol. 521, pp. 436-441, 2015.

[13] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning*, Beijing, China, 2014.

[14] R. Jia, M. Jin, K. Sun, T. Hong and C. Spanos, "Advanced building control via deep reinforcement learning," in *10th International Conference on Applied Energy* , Hong Kong, China, 2018.

[15] Y. Li, Y. Wen, K. Guan and D. Tao, "Transforming cooling optimization for green data center via deep reinforcement learning," *arXiv.org,* 2018.

[16] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Applied Energy,* vol. 235, pp. 1072-1089, 2019.

[17] R. van Emden and M. Kaptein, "contextual: Evaluating Contextual Multi-Armed Bandit Problems in R," *arXiv.org,* 2018.

[18] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv.org, 2017.

[19] K. He, X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-levelperformance on ImageNet classification," arXiv.org, pp. 1-11, 2015.