Reinforcement Learning Based Load Balancing for Geographically Distributed Data centres

Max Mackie, Hongjian Sun
Department of Engineering
Durham University,
Durham, DH1 3LE, U.K.
Emails: {max.mackie, hongjian.sun}@durham.ac.uk

Jing Jiang
Department of Mathematics, Physics &
Electrical Engineering, Northumbria University,
Newcastle, NE1 8ST, U.K.
Email: jing.jiang@northumbria.ac.uk

***Abstract*—This paper proposes a method of migrating workload among geo-distributed data centres that are equipped with on-site renewable energy sources (RES), such as solar and wind energy, to decarbonise data centres. It aims to optimise the performance of such a system by introducing a tunable Reinforcement Learning (RL) based load-balancing algorithm that implements a Neural Network to intelligently migrate workload. By migrating workload within the network of geo-distributed data centres, spatial variations in electricity price and intermittent RES can be capitalised upon to enhance data centres' operations. The proposed algorithm is evaluated by running simulations using real-world data traces. It is found that the proposed algorithm is able to reduce costs by 6.1% whilst also increasing the utilisation of RES by 10.7%.**

***Index Terms*—Data centre, Machine Learning, Load Balancing, Reinforcement Learning**

I. Introduction

THE rapid development of the digital economy has led to exponential growth in the number of data centres. Large service providers operate many data centres, each powering and cooling thousands of servers which consume vast amounts of power [1]. Data centres' energy consumption is projected to be 4.5% of global energy use by 2025 [2], leading to significant CO_2 emissions. This has driven the adoption of renewable energy sources (RESs), such as wind or solar energy, as a promising solution for companies aiming to reduce their CO_2 emissions due to either legislation requirements or increasing public scrutiny. Since a large proportion of a data centre's load is made up of delay-tolerant tasks, existing studies have mainly concentrated on how to intelligently dispatch and schedule them to increase penetration of RES and minimise costs.

Kwon [3] considered the co-location of solar power generation with large-scale battery storage for improving RES utilisation by using a mathematical optimisation model that successfully minimised energy cost whilst ensuring the desired level of RES with the required service guarantee quality. Li *et al.* [4] reduced data centres' dependence on large scale battery storage systems by applying an I-switch algorithm to integrate RES. It intelligently shifts the computing load from one energy source to another to best achieve load power matching.

To deal with the intermittency of RES, Buyya *et al.* [5] proposed a short-term prediction algorithm using a Gaussian mixture model. The algorithm trained itself by using previously observed energy levels to predict future time steps' energy level. Goiri *et al.* [6] developed a parallel batch job scheduler that used short-term solar predictions to make effective scheduling decisions depending on job deadlines. Alsanli *et al.* [7] studied how to use prediction models to allocate load between geo-distributed data centres whilst considering the impact of wide-area networks on data centres.

Lei *et al.* [8] focused on maximising the utilisation of RES and customer satisfaction whilst minimising total cost by using an enhanced co-evolutionary algorithm. Deng *et al.* [9] developed an online power management and load scheduling algorithm which models the problem as a constrained stochastic optimisation problem. Qi *et al.* [10] furthered this research by developing a new Lyapunov optimisation-based algorithm that utilised task scheduling to minimise costs by deferring delay-tolerant tasks to periods with reduced carbon emission rates. Toosi *et al.* [11] aimed to exploit the spatiotemporal variations in on-site power and grid prices by balancing data centre load among numerous geo-distributed data centres.

Zhou *et al.* [12] developed a Reinforcement Learning (RL) based adaptive resource management algorithm that aims to balance power consumption and quality of service revenue. Xu *et al.* [13] studied a neural network to evaluate the RL algorithm's value function and applied it to the minimisation problem of big data analytics on geo-distributed data centres. They successfully developed an algorithm that used performance-enhancing techniques, such as random pool sampling and unidirectional bridge network structures, to reduce computational complexity.

Compared to the algorithm proposed in [13], the unique contribution of this paper is to produce an RL based load scheduling algorithm for geo-distributed data centres using a tunable value function. Such a value function can help data centre operators adapt to political or economic changes, e.g., the UK government's net-zero target or the introduction of a carbon tax. The implementation of a neural network to estimate a tunable value function has not yet been researched in existing studies.

The remainder of this paper is organised as follows: Section II discusses the problem definition. Section III presents the proposed RL based algorithm. The simulation results are described in section IV. Finally, Section V concludes the paper and discusses potential areas for future research.

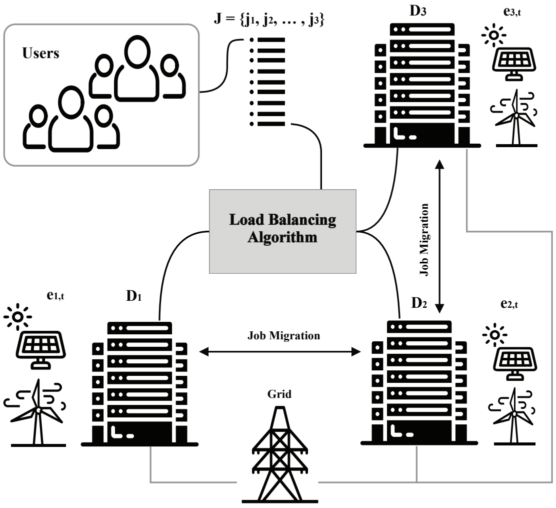


Fig. 1: System Model

II. System Model

In this section, a model similar to that adopted in [13] is employed. It explores three geographically distributed data centres $D = \{d_1, d_2, d_3\}$ each partially powered with on-site solar and/or wind energy generations. The model considers a large set of job requests $J = \{j_1,\ j_2 \dots j_m \dots j_M\}$ arriving at different gateways which then pass them on to three geographically distributed data centres. The power consumption of each data centre is dependent on the data arrival rate, c_k of job j_k. When this data rate is high, the data centre will have to increase the number of active servers leading to higher power consumption. The CPU utilisation of the i-th data centre is denoted by r_i^c. The total energy consumption $u_{i,t}$ at time t is calculated using a discrete-time power model with T time intervals

$$u_{i,t} = \int_{t_0}^{t} \left[\sum_{d_i \in D} P_i\left(t\right) \right] dt, \tag{1}$$

where

$$P_i\left(t\right) = p_i^{idle} + \left(p_i^{full} - \ p_i^{idle}\right)\left(2r_i^c\left(t\right) - \{r_i^c(t)\}^{1.4}\right), \tag{2}$$

where p_i^{idle} and p_i^{full} represent the power of data centre d_i when the CPU utilisation is at 0% and 100% utilisation respectively. Server provisioning and other power management techniques allow server systems to reduce consumption when the CPU is not fully utilised. A typical server will consume 40% of its peak power when idle [14], so for this paper $p_i^{idle} = 0.4 p_i^{full}$.

As shown in [15], available on-site wind power can be modelled as a function of the actual wind speed at each data centres location. The power output of a single turbine P_{wind} for a given wind speed $v\left(t\right)$ can be approximated as follows

$$P_{wind} = \begin{cases} 0, & v\left(t\right) \le v_{in}\ ,\quad v\left(t\right) \ge v_{out} \\ P_r\left(\frac{v(t)\ -\ v_{in}}{v_r - v_{in}}\right), & v_{in} < v\left(t\right) < v_r \\ P_r, & v_r \le v\left(t\right) < v_{out} \end{cases} \tag{3}$$

where v_{in} and v_{out} are the turbine cut-in and cut-out speeds, respectively. The total power P_W generated from X wind turbines is given by

$$P_W = \sum_{x=1}^{X} P_{wind}^{x}. \tag{4}$$

The available on-site solar power is modelled as a function of solar irradiance. The output power of a photovoltaic panel is given by

$$P_{solar} = \alpha \cdot A \cdot s\left(t\right) \cdot du \tag{5}$$

where α is the efficiency of electrical conversion, A represents the total solar panel area, $s(t)$ is the solar irradiance, and du is the time duration of solar irradiance.

To reduce carbon emission, we prioritise renewable energy over brown energy, in other words, the grid electricity is only required when all available renewable energy is consumed. The energy cost at time slot t is calculated using,

$$C_{en}^{t} = \rho_{i,t}^{brown} \cdot \max\left(u_{i,t} - e_{i,t}, 0\right) + \rho_{i,t}^{green} \cdot \min\left(u_{i,t}, e_{i,t}\right) \tag{6}$$

where $e_{i,t}$ is the available renewable energy at time t, and the price of renewable and brown energy of the i-th data centre at time t, are denoted as $\rho_{i,t}^{green}$ and $\rho_{i,t}^{brown}$ respectively.

The impact of job migration at time t can be given by using core relevant factors [13],

$$C_{mig}^{t} = \sum_{j_k \epsilon J} M_{j_k} \cdot L(d, d') \cdot \sigma_{j_k} \tag{7}$$

where M_{j_k} represents the memory size of the job being migrated, $L(d, d')$ represents the distance from current data centre d to destination data centre d', and σ_{j_k}represents the sensitivity to migration of the job j_k.

A trade off between energy cost and impact of job migration can be described by

$$C \ = \ \sum_{t\ =1}^{T} \left(\beta \cdot C_{en}^{t} + (1-\beta) \cdot C_{mig}^{t}\right), \tag{8}$$

where β allows the cost of energy and the job migration impact to be varied. This allows for data centre operators to customise the value function so that the tradeoff between RES utilisation and migration impact can be balanced to meet specific data centre requirements.

III. Load Balancing Algorithm Design

The algorithm proposed in this paper is designed to balance load by migrating jobs between geo-distributed data centres using feedback from historical migration decisions. Each interaction begins with the agent selecting an action and interacting with the environment by informing the data centre. The data centre then executes this action by either processing the job

Algorithm 1 Load Balancing Algorithm

```
begin
Initialise neural network $F(s_t, a_t)$ with random weight $w$ and bias $b$.
for  time step $t \in (1, 2, ..., T)$ do
    Update available green power at each data centre.
    for  job identity $m \in (1, 2, ..., M)$ do
        Generate random number $z_m \in [0, 1]$
        if $z_m < \eta$  then
            Select the minimum of neural network output
        else
            Randomly select an action $a_t^*$.
        end if
        Execute $a_t^*$, update $r_{t+1}$ and $s_{t+1}$.
        Add dataset ($a_t^*$, $r_{t+1}$, $s_{t+1}$) to batch.
        Set $s_t = s_{t+1}$.
    end for
end for
end
```

TABLE I: Experimental Setup of RL Algorithm

Parameters	
$\lambda = 0.5$	$p_i^{idle} = 0.4 p_i^{full}$
$\eta = 0.9$	$\beta = 0.5$
$BatchSize = 100$	$NeuralNetwork = B$

at its current location or migrating it to one other data centre locations. The agent then receives feedback and the next state from the environment. At the start of each time step t, the load balancer chooses an action a_t for each job, where the action corresponds to the location selected to complete the job. These are chosen using information contained in the current state of the system s_t. Once all of these jobs are sent to the corresponding data centres, the load balancing agent observes the new state of the system s_{t+1} and the reward $r_{t+1} = \beta C_{en}^t + (1 - \beta) C_{mig}^t$ before transitioning into the next time step.

The load balancing algorithm's pseudo-code is provided in **Algorithm 1**. This proposed algorithm begins by initialising a neural network that takes s_t as the input and uses it to approximate the reward associated with each possible action. The optimal action can then be found by taking the minimum value of the neural network output.

At each time step, it starts by calculating the available solar and wind power for each data centre using weather data from the environment's state. The algorithm then generates a random number, $z \epsilon [0, 1]$ for each job and compares this with η to determine how the selected action a_t^*, is chosen. If a job's z is less than an exploration parameter η, then the minimum value of the neural network output is selected. Otherwise a_t^* is chosen randomly to allow for exploration of the environment's state-space. The action a_t^* is then executed and the algorithm moves to the next job in the time step. At the end of each time step, the environment transitions to the next state s_{t+1} and the associated reward r_{t+1} is added to the batch of historical results. If this batch is full, then the data is shuffled and used to retrain the neural network using Stochastic Gradient Descent. Stochastic Gradient Descent methods attempt to minimize the error by slightly shifting the neurons weight vector in the direction that minimizes the error for that set of examples [16]. By repeating the retraining process, the neural network could converge towards an optimal load balancing policy.

The algorithm's performance can be evaluated by considering its time complexity, which is a measure of the number of elementary operations performed by the algorithm. Since the retraining process dominates the proposed algorithm's runtime, it is used to approximate the time complexity. The retraining process uses Stochastic Gradient Descent, which relies on back-propagation. The time complexity of back-propagation in a fully connected multilayer perception type neural network can be approximated by $O(I^2 \cdot La)$ where I is the number of nodes in the hidden layer, and La is the number of layers. The time complexity of the whole retraining process is thus,

$$\text{Time} \sim O(T \cdot B \cdot I^2 \cdot La) \tag{9}$$

where T is the number of iterations, and B is the number of training samples in each batch. The actual runtime of the algorithm will be far smaller when implemented in reality. Matrix multiplication can be accelerated using mathematical techniques such as the Strassen algorithm or the Coppersmith-Winograd algorithm, which can approximate matrix multiplication in $O(n^{2.375})$ and $O(n^{2.807})$ respectively which is much faster than the $O(n^3)$ associated with basic matrix multiplication. The algorithm runtime can also be reduced considerably using parallel processing, which can be easily implemented in data centre operation.

IV. Experiments and Results

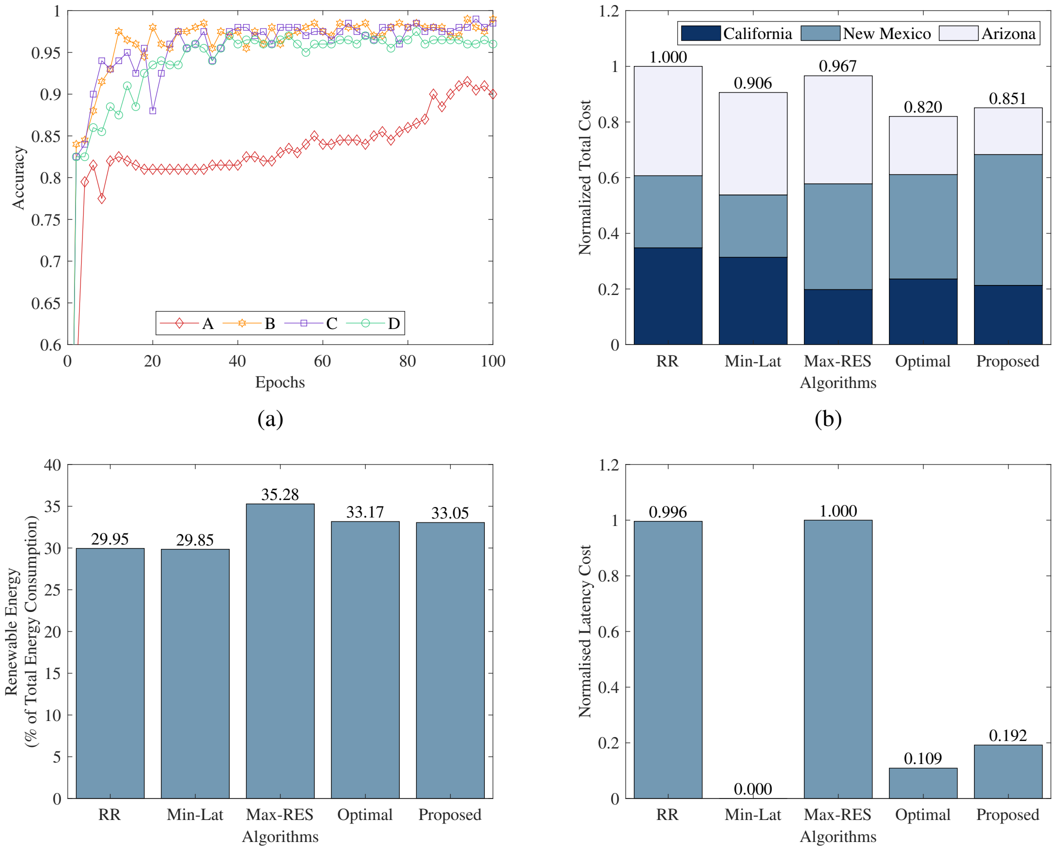
This section evaluates the proposed algorithm based on real-world traces of user workloads, electricity prices, and available on-site wind and solar power. The experimental setup for the simulations will first be described, and then the results will be presented along with detailed analysis.

A. Experimental Setup

To evaluate the performance of the algorithm, the following datasets were used,

1) ***Available renewable energy:*** To find the available wind and solar energy, weather data was used for three potential data centres located in Lancaster, California (CA), Flagstaff, Arizona (AZ) and Santa Rosa, New Mexico (NM). It was sourced from [17] and [18].
2) ***Electricity price from grid:*** Prices for the three data centre locations were sourced from the Energy Information Administration [19].
3) ***Data centre workload:*** The Google cluster data-trace represents 29 days' worth of Borg cell information from May 2011 on a cluster of about 12.5k machines [20]. Each job has a scheduling class.

It was found that the exploration parameter η clearly influenced the performance. This is because, at low η, the algorithm prioritises exploration over exploitation. The



(a)

(b)

(c)

(d)

Fig. 2: Performance comparisons: (a) Different neural network structures on learning rates, (b) Cost, (c) RES utilisation, and (d) Latency cost, of different algorithms.

TABLE II: Neural Network Configurations

ID	No. of Layers	Structure (Neurons/Layer)
A	4	9-20-10-3
B	5	9-45-30-15-3
C	8	9-45-40-35-30-20-10-3
D	10	9-150-115-90-50-30-20-15-10-3

exploration-exploitation dilemma is where an RL algorithm must exploit knowledge gained from previous experiences to achieve reward; however, it also must explore the environment so that forthcoming decisions are better informed. Xu's paper [13] found that the algorithm performed optimally for higher values of η. Therefore, the exploration parameter was set at $\eta = 0.9$. For the neural network itself, a multilayer perception type neural network with five layers was chosen.

B. Performance comparisons

To better understand the learning rate, Figure 2(a) shows the accuracy of different neural network configurations shown in Table II. It can be seen that neural network B converges fastest, closely followed by C. Structure A converges slowly and clearly underfits the problem. In contrast, D overfits the training data and does not achieve the same accuracy as B and C when tested on validation data. For the proposed algorithm, structure B was chosen as it offered high accuracy without adding unnecessary complexity. As can be seen from the computational complexity described above, the number of neural network layers significantly affects the algorithm's overall runtime.

We also compare the proposed algorithm with four benchmark algorithms in terms of total cost, migration cost and renewable energy utilisation.

1) ***Round-Robin (RR):*** This algorithm schedules jobs to data centres in a round-robin manner without any information about RES [21]. Although the RR algorithm is simple, its performance can be used to exhibit the importance of the migration policy for minimising data centre costs and improving RES utilisation. RR is unable to exploit spatial variations in the available renewable energy or electricity price to reduce total cost. It also incurs high migration costs as it has no knowledge of job sensitivity, job size or migration distance.
2) ***Minimum Latency (Min-Lat):*** This algorithm aims to minimise the job latency by always completing each task

at the data centre that it was initially scheduled to avoid ever migrating tasks between data centres. Min-Lat is used to evaluate the performance of a data centre without workload migration. Similarly to RR, this algorithm cannot utilise spatial variations; however, it does not incur migration costs.

3) ***Maximum RES (Max-RES):*** This algorithm aims to maximise RES utilisation by always completing each task at the data centre with the smallest variation between the available RES and current energy usage [22]. Max-RES can fully utilise the spatial variations in electricity price and RES availability. Its performance can be used to show the optimal utilisation of renewable energy that the proposed algorithm will attempt to achieve. However, without knowledge of job characteristics, this algorithm will incur high migration costs similar to RR.
4) ***Offline Optimal Solution (Optimal):*** This algorithm represents the optimal solution of the load balancing problem. The neural network is trained with complete future information of system states thus incurring high computational cost. The offline algorithm requires all data at the beginning of the simulation so is not applicable to real-time load balancing problems; however, it does offer a good metric for evaluating the performance of the proposed online algorithm.
5) ***Proposed RL-Based Algorithm (Proposed)*** This is the proposed online algorithm detailed in Section III.

Figure 2(b) shows the total cost of the simulation using each of the benchmark algorithms. The results are normalised to the maximum cost, which was incurred using the RR algorithm. The graph shows that the proposed algorithm can minimise total cost compared to RR, Min-Lat and Max-RES. The proposed algorithm shows a 6.1% cost reduction when compared to Min-Lat. Since the Min-Lat algorithm represents data centre performance without workload migration, this reduction can be attributed to intelligent migration of jobs, better using available renewable energy and variations in electricity price. It is worth noting that the cost reduction was achieved while incurring a significant migration cost which was not incurred by the Min-Lat algorithm. This can be seen in Figure 2(d) which shows the normalised migration cost associated with each algorithm. Evidence for the proposed algorithm's opportunistic migration of jobs can be seen in Figure 2(c) which shows the percentage of total energy consumption supplied by RES. The proposed algorithm boasts a 10.7% increase in RES utilisation when compared to the Min-Lat algorithm. The Max-RES algorithm achieved the optimal utilisation of RES; however, this was reached by disregarding job sensitivity, job size and migration distance, resulting in a migration cost over five times that incurred by the proposed algorithm.

When comparing the total costs of the offline optimal algorithm against the proposed algorithm in Figure 2(b); the proposed algorithm is able to achieve 82.7% of the potential cost savings demonstrated by the optimal algorithm when taking RR as the reference. In terms of RES utilisation, the proposed algorithm shows only a slight reduction from the optimal algorithm suggesting that the it can correctly identify potential savings due to variations in available RES. When comparing the proposed algorithms migration costs with the offline optimal, it shows a relatively large increase of 76%. This results from algorithm's exploration parameter $\eta = 0.9$ which results in 10% of actions being chosen randomly.

C. Sensitivity analysis

This section evaluates the effects of varying the value function tuning parameter β on the performance of proposed algorithm. Figure 3(a) and Figure 3(b) shows the impact that varying β have on the migration cost and renewable energy utilisation. It can be seen that increasing β reduces migration costs at the expense of renewable energy utilisation. Therefore, the tunable parameter integrated into the RL value function gives data centre operators the ability to vary the level of task migration to balance latency and utilisation requirements.

Figure 3(c) compares the RES utilisation of Max-RES, Min-Lat, and the proposed algorithm as the amount of available RES is varied. As can be seen, the proposed algorithm can outperform the Min-Lat algorithm for all RES capacity values. The Max-RES represents the upper bound of potential RES utilisation achievable with job migration. The proposed algorithm can fulfil a significant proportion of this potential, even as the RES is increased. This is an essential factor in the future-proofing of this load balancing approach as the penetration of renewable energy will increase over the coming years to meet emission reductions targets set out in political legislation such as the 2015 Paris agreement.

V. Conclusion and Future Work

To reduce carbon emissions and electricity cost, many service providers have built data centres with on-site renewable energy generation. Larger companies tend to have a network of geographically distributed data centres, which allows for potential cost savings through the migration of workload to take advantage of spatial variations in renewable energy availability. However, due to the complex nature of data centre operation, load-balancing optimisation is a challenging problem to solve. This paper investigates a tunable RL based load balancing algorithm that integrates a neural network into the RL framework to approximate the optimal solution with minimum cost. Extensive simulations were run using real-world data traces of user workload, available renewable energy and grid prices. These show that the proposed algorithm can reduce the total cost of the data centre by 6.1% compared to the Min-Lat algorithm. The proposed algorithm can also increase the utilisation of RES by 10.7%.

To build upon this paper, there are several directions that could be further researched. One aspect worth exploring is the RL-based method's performance as the distributed data centre network size increases. Another aspect that should be explored is improving the proposed model to optimise task scheduling in heterogenous data centre environments by considering more constraints associated with real data centre operation. For

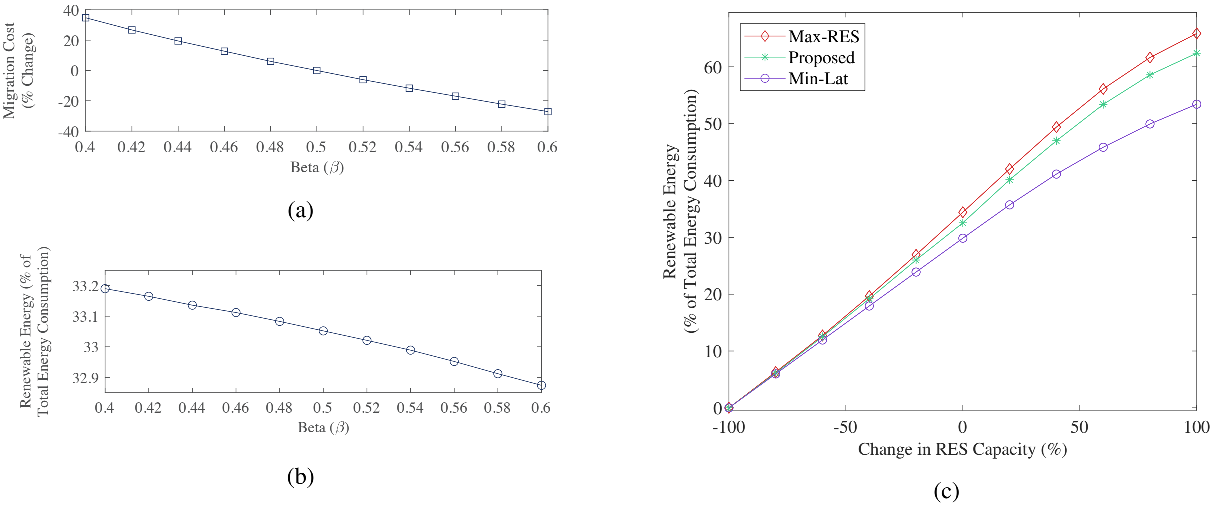


Fig. 3: (a)Variation of migration cost with β, (b) Variation of renewable energy utilisation, and (c) Variation of renewable energy utilisation with RES capacity.

example, it was assumed in this paper that all virtual machines' ability was the same and that each could only complete one task at a time. These further explorations could build on the findings of this paper helping to further demonstrate the effectiveness of RL-based algorithms for optimising data centre workload scheduling.

References

[1] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.

[2] I.T. Union, "ICT statistics kernel description," 2020, [Accessed 20 Dec 2020]. [Online]. Available: https://www.itu.int/en/ITUD/Statistics/Pages/stat/default.aspx

[3] S. Kwon, "Ensuring renewable energy utilization with quality of service guarantee for energy-efficient data center operations," *Applied Energy*, vol. 276, p. 115424, 2020.

[4] C. Li, A. Qouneh, and T. Li, "iSwitch: Coordinating and optimizing renewable energy powered server clusters," *ACM SIGARCH Computer Architecture News*, vol. 40, no. 3, pp. 512–523, 2012.

[5] A. Khosravi and R. Buyya, "Short-term prediction model to maximize renewable energy usage in cloud data centers," in *Sustainable Cloud and Energy Services*. Springer, 2018, pp. 203–218.

[6] Í. Goiri, K. Le, M. E. Haque, R. Beauchea, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Greenslot: scheduling energy consumption in green datacenters," in *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, pp. 1–11.

[7] B. Aksanli, J. Venkatesh, I. Monga, and T. S. Rosing, "Renewable energy prediction for improved utilization and efficiency in datacenters and backbone networks," in *Computational Sustainability*. Springer, 2016, pp. 47–74.

[8] H. Lei, R. Wang, T. Zhang, Y. Liu, and Y. Zha, "A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center," *Computers & Operations Research*, vol. 75, pp. 103–117, 2016.

[9] Geeksforgeeks.com, "Online algorithm," 2010, [Accessed 20 Dec 2020]. [Online]. Available: https://www.geeksforgeeks.org/online-algorithm/

[10] H. Dou, Y. Qi, W. Wei, and H. Song, "Carbon-aware electricity cost minimization for sustainable data centers," *IEEE Trans. Sustainable Computing*, vol. 2, no. 2, pp. 211–223, 2017.

[11] A. N. Toosi, C. Qu, M. D. de Assunção, and R. Buyya, "Renewable-aware geographical load balancing of web applications for sustainable data centers," *Journal of Network and Computer Applications*, vol. 83, pp. 155–168, 2017.

[12] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *Proc. 2017 IEEE/ACM 25th International Symposium on Quality of Service*. IEEE, 2017, pp. 1–6.

[13] C. Xu, K. Wang, P. Li, R. Xia, S. Guo, and M. Guo, "Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning," *IEEE Trans. Network Science and Engineering*, vol. 7, no. 1, pp. 205–215, 2018.

[14] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH computer architecture news*, vol. 35, no. 2, pp. 13–23, 2007.

[15] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Proc. International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 2011, pp. 143–164.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[17] Solecast, "Data and specifications," 2020, [Accessed 15 Dec 2020]. [Online]. Available: https://solcast.com/historical-and-tmy/data-and-specifications/

[18] National Oceanic and Atmospheric Administration, "Weather," 2020, [Accessed 17 Dec 2020]. [Online]. Available: https://www.noaa.gov/weather

[19] U.S. Energy Information Administration, "Wholesale electricity and natural gas market data," 2011, [Accessed 15 Dec 2020]. [Online]. Available: www.eia.gov/electricity/wholesale/

[20] J. Wilkes, "Google cluster data," 2011, (Accessed 16 Nov 2020).

[21] A. Toosi and R. Buyya, "A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers," in *Proc. 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2015.

[22] C. Chen, B. He, and X. Tang, "Green-aware workload scheduling in geographically distributed data centers," in *Proc. 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, 2012, pp. 82–89.