Energy and AI 9 (2022) 100179

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Reinforcement learning and A* search for the unit commitment problem

Patrick de Mars *, Aidan O'Sullivan

UCL Energy Institute, United Kingdom

ARTICLE INFO

Keywords:
Unit commitment
Reinforcement learning
Tree search
Power systems

ABSTRACT

Previous research has combined model-free reinforcement learning with model-based tree search methods to solve the unit commitment problem with stochastic demand and renewables generation. This approach was limited to shallow search depths and suffered from significant variability in run time across problem instances with varying complexity. To mitigate these issues, we extend this methodology to more advanced search algorithms based on A* search. First, we develop a problem-specific heuristic based on priority list unit commitment methods and apply this in Guided A* search, reducing run time by up to 94% with negligible impact on operating costs. In addition, we address the run time variability issue by employing a novel anytime algorithm, Guided IDA*, replacing the fixed search depth parameter with a time budget constraint. We show that Guided IDA* mitigates the run time variability of previous guided tree search algorithms and enables further operating cost reductions of up to 1%.

1. Introduction

The unit commitment (UC) problem is one of the fundamental decision-making problems faced by power system operators and generating companies. The task is to determine the on/off schedules of thermal generating units to meet demand at minimum cost, while respecting generator operating constraints and allocating sufficient reserve capacity to manage contingencies [1]. To participate in forward or day-ahead power markets and allow sufficient time for system operators to conduct system security analysis, this optimisation problem must be solved hours or days in advance of delivery [2]. As a result, solutions to the UC problem must take into account uncertainties arising from demand, renewables generation and outages of generation or transmission assets. With rising penetrations of wind and solar PV generation, the variability of generation is becoming an increasingly important consideration for system operators, requiring new methodologies for producing economic and reliable UC solutions [3]. In this paper, we describe and test two new algorithms for solving the UC problem, based on model-free reinforcement learning (RL) and model-based tree search.

1.1. Motivation

The industry standard in UC solution methods is mixed-integer linear programming (MILP) [2]. However, MILP is not a natural framework for accounting for uncertainties and requires the use of reserve constraints based on expert heuristics and rules of thumb. RL is a promising methodological framework for solving the UC problem which can be used to learn optimal control strategies in stochastic environments [4]. An additional benefit of RL is the ability to shift the majority of the computational burden offline to a training phase. Combining RL with tree search has emerged in the artificial intelligence (AI) literature as a powerful methodology, and is the state of the art in widely-studied games-playing domains [5,6].

In previous work, an RL-aided tree search approach combining model-free and model-based methods was used to solve the UC problem with stochastic demand and renewables generation [7]. This *guided tree search* algorithm used a policy trained with model-free RL to intelligently reduce the branching factor of a search tree. The reduced search tree was solved with the general-purpose algorithm uniform-cost search (UCS), a variant of Dijkstra's algorithm [8]. Guided UCS outperformed conventional deterministic UC approaches using MILP on 20 problem instances, reducing total operating costs by 0.3–0.9% [9].

The search depth H of Guided UCS was limited to 4 timesteps or a 2-hour lookahead horizon. Increasing the search depth results in greater foresight and lower operating costs; deeper search was not possible due to exponential time complexity in H [7]. In addition, when applied to problem instances of differing complexity, the run time of Guided UCS was found to vary by an order of magnitude. Shallow search depth and run time variability are significant limitations of Guided UCS which are addressed in this research.

The results of [7] indicate that combining RL with tree search methods is an effective methodology for solving the UC problem that

* Corresponding author.
E-mail address: patrick.demars@ucl.ac.uk (P. de Mars).

https://doi.org/10.1016/j.egyai.2022.100179
Received 9 March 2022; Received in revised form 27 May 2022; Accepted 29 June 2022
Available online 2 July 2022

is competitive with conventional mathematical optimisation methods. Motivated by these promising results, this paper makes significant improvements on Guided UCS by adopting more advanced search methods based on A* search.

1.2. Contributions

To mitigate the issues of limited search depth and run time variability in Guided UCS [7], in this paper we extend the guided tree search methodology to problem-specific search methods: A* search [10] and iterative-deepening A* search [11].

Both methods employ a problem-specific heuristic based on priority list (PL) UC methods [12], which approximates the value of nodes in the search tree. The novel PL algorithm developed in this research orders generators by their marginal fuel costs, and commits them while relaxing constraints to rapidly produce a 'best-case' schedule which is used to identify promising branches of the search tree. In experiments considering power systems of up to 30 generators, we show that the heuristic improves search efficiency when applied in Guided A*; total run times are reduced by up 94% as compared with the heuristic-free Guided UCS algorithm, with negligible impact on operating costs.

The second algorithm, Guided IDA* search, exhibits practical advantages in power system contexts as it is *anytime*, with the search depth parameter H replaced by a time budget constraint b. This mitigates the large run time variability of existing UC solution methods across problems instances of varying complexity and enables greater search depths to be reached without risk of prohibitively long computing times. For similar computational budgets, Guided IDA* results in operating cost savings of up to 1% as compared with Guided UCS. Guided IDA* is capable of solving diverse problem instances reliably within limited computation times, an essential and under-researched property of RL solution methods for the UC problem.

This paper provides further evidence that RL-aided (guided) tree search is a powerful methodology for solving the UC problem and shows that performance improvements can be achieved through exploiting knowledge of power systems, motivating further collaboration between domain experts and RL practitioners.

In summary, this paper makes the following contributions:

- We introduce two new guided tree search algorithms for the UC problem, Guided A* search and Guided IDA* search, applying the principle of guided expansion introduced in [7]. Both algorithms are informed, using a problem-specific heuristic to improve search efficiency.
- A heuristic based on a PL algorithm is introduced for application in Guided A* and Guided IDA*. The heuristic is analysed in terms of average run time, accuracy and admissibility and applied in Guided A* search to solve UC problem instances of up to 30 generators with stochastic demand and wind generation. We evaluate improvements to search efficiency improvements as compared with previous work [7], finding mean run time is reduced by between 64%–94% as compared with Guided UCS, without significant changes in operating costs.
- The anytime algorithm Guided IDA* search is applied to solve UC problem instances and found to allow for deeper search on average while minimising run time variability. Operating costs are found to be between 0.4–1.0% lower than Guided UCS while completing in similar run time.

1.3. Article structure

In the following section, we conduct a literature review of UC solution methods with focus on RL-based approaches. Section 3 describes the problem setup and RL environment adopted in our experiments. In Section 4 we describe Guided A* and Guided IDA* algorithms. Section 5 describes experiments applying Guided A* and Guided IDA* to solve UC problem instances with up to 30 generators. We discuss the results in Section 6 and Section 7 concludes the paper.

2. Literature review

In this section we briefly present key results of research using mathematical optimisation techniques to solve the UC problem before reviewing the state of the art in RL for UC. For more complete reviews of the broader UC literature, see [2,13].

2.1. Mathematical optimisation for unit commitment

A large body of research has been dedicated to solving the UC problem with optimisation methods including MILP [2,14], Lagrangian relaxation [15,16] and metaheuristic methods such as genetic algorithms [17], particle swarm optimisation [18] and simulated annealing [19]. The UC problem is usually framed as a deterministic optimisation problem with uncertainties managed by enforcing a reserve constraint, requiring that excess capacity is committed to manage deviations in demand and renewables generation from their forecasts and other contingencies such as outages [14]. Reserve constraints are typically determined using heuristic methods, such as the widely used $N-1$ criterion protecting against the single largest loss of infeed [20] or criteria based on the distribution of forecast errors [21]. MILP is the dominant methodology for solving the UC problem in practical contexts [2], and its adoption is estimated to have resulted in annual operating cost savings of $150 million per year in the PJM interconnection alone as compared with previous Lagrangian relaxation methods [22].

Scenario-based stochastic optimisation methods have been widely-studied and shown to achieve lower expected operating costs than deterministic UC [23–25]. The magnitude of cost improvements are 0.25–0.9% in a study of the 2020 Irish power system [25], and 1.3% lower in experiments on the IEEE RTS system of 32 generators [23]. However, a significant drawback of stochastic optimisation methods is their much larger computational expense [26]; run times of stochastic UC with 12 scenarios and reserve requirements are found to be between 1 and 3 orders of magnitude larger than deterministic UC in [23]. This has motivated further research into new solution methods which more rigorously account for uncertainty while remaining computationally tractable in short computing times.

2.2. Reinforcement learning for unit commitment

RL has been recognised as a promising framework for solving the UC problem [33–35] but makes up only a small fraction of the existing UC literature. RL has previously been used to solve the UC problem in [7,27–32,36], which are summarised in Table 1. Most studies have used model-free RL, notably Q-learning [27–30,32], which has been applied to solve UC problems of up to 10 generators. All of these studies except [32] considered training and testing on a single episode; as a result, they do not demonstrate the ability of the trained policy to generalise to unseen problems – an important advantage of RL over other optimisation methods – or the variation in solution time and quality for days of varying complexity. Only two of the Q-learning studies include uncertainty in the problem setup through stochastic renewables generation [28,29]. Fuzzy Q-learning is used in [30] to solve the widely-studied Kazarlis et al. benchmark problem with 10 generators, and is shown to outperform several existing deterministic UC solution methods. An adapted multi-step deep Q-learning algorithm is used to solve UC problems with deterministic load and 5 generators, with comparable results to mixed-integer quadratic programming in [32]. This is the only study reviewed which trained on multiple days and evaluated final performance on a held-out set. The Q-learning methods studied suffer from curses of dimensionality in the state and action spaces for the UC problem, which has limited applications to systems of up to 10 generators.

A larger study of 99 generators is studied in [36], where the UC problem and real-time dispatch are represented in an interleaved

Table 1
Summary of research applying RL to the UC problem. We show the method used; the maximum problem size by number of generators; if experiments involved stochastic demand or generation; if multiple days were used in training; if performance was evaluated on unseen test days.

Study	Method	Gens.	Stochastic setup	Multiple training days	Unseen test days
Jasmin et al. 2009 [27]	Q-learning	4	No	No	No
Jasmin et al. 2016 [28]	Q-learning	10	Yes	No	No
Li et al. 2019 [29]	Q-learning	10	Yes	No	No
Navin & Sharma [30]	Multi-agent Q-learning	10	No	No	No
Dalal & Mannor, 2015 [31]	SARSA	8	No	No	No
Dalal & Mannor, 2015 [31]	Tree search	12	No	No	No
Qin et al. 2021 [32]	Q-learning	5	No	Yes	Yes
de Mars & O'Sullivan, 2021 [7]	Guided tree search	30	Yes	Yes	Yes

Markov decision process (MDP) and solved with the cross entropy method. However, the UC component of this problem is simplified significantly to selecting a single commitment decision for each 24-hour period, with no intra-day commitment changes, and the action space is significantly reduced to 20 actions. As a result, this study cannot be compared directly with other UC research including our own.

Model-based methods based on tree search have been applied to solve the UC problem in [7,31]. Model-based methods offer advantages over model-free methods in contexts of critical infrastructure such as power systems operation as lookahead strategies are employed to improve robustness of solutions [37]. Tree-based methods without RL are used to solve a deterministic UC problem instance of 12 generators in [31], and found to outperform a metaheuristic solution by 27% in terms of operating costs. The problem setup used did not consider stochastic demand or wind generation or evaluate performance and run time in generalising across multiple problem instances. Guided tree search, which combines model-free RL with tree search and is the basis of this research, was developed in [7] and applied to stochastic problem instances with up to 30 generators. We describe guided tree search in more detail in Section 4. Proximal policy optimisation (PPO) was used for training and uniform-cost search (UCS) used to solve 20 unseen profiles. Compared with a deterministic UC benchmark solved with MILP, operating costs were reduced by between 0.3–0.9%. The run time of UCS without RL enhancement was shown to grow exponentially in the number of generators, while using Guided UCS the computational cost remained roughly constant [7]. However, run times were highly variable across problem instances of the same number of generators using both UCS and Guided UCS, which raises practical challenges in time-constrained contexts. An ideal RL method for the UC problem should reliably produce high quality solutions in practical run times across problem instances with different characteristics. Furthermore, UCS is a simple, general-purpose search algorithms, and performance can be improved by developing problem-specific methodologies for the UC problem. The two algorithms developed in Section 4, Guided A* and Guided IDA*, address these shortcomings of Guided UCS, exhibiting improved search efficiency and reduced operating costs. The following section describes the problem setup adopted in our experiments.

3. Problem setup

In this section we describe the problem setup used in Section 5 to evaluate the performance of guided tree search algorithms across UC problem instances with uncertain demand and renewables generation. The problem setup employed is identical to [7], allowing for direct comparison of our results. We briefly describe the RL environment used to train and evaluate the novel guided tree search algorithms and the formalisation of the problem as an MDP.

3.1. Power system environment

In order to apply RL to the UC problem, a power system simulation environment is required to enable training of RL agents by trial-and-error. Previous work described a power system environment for the UC problem[1] which is designed to emulate day-ahead UC decision-making given forecasts for demand and wind generation [7]. We adopt the environment in this research and provide a brief description here; the problem is then formalised as an MDP in Section 3.2. The environment models a power system of N generators with 48 30-minute settlement periods per day, reflecting the GB power market structure. The generators are specified using data from a widely-used benchmark [17], which gives quadratic fuel cost curves, start costs, and minimum up/down time constraints for 10 generators. For each day, forecasts for demand and wind generation are specified based on historical data from the GB power system and the environment follows the routine shown in Fig. 1. The environment processes each commitment decision (action in the MDP) sequentially and samples forecast errors, which are represented by auto-regressive moving average (ARMA) processes. The operating costs are calculated by solving the economic dispatch (ED) problem with the lambda-iteration method [1] to determine the real-valued power outputs of generators required to meet the demand net of wind generation. If the net demand cannot be met, then the volume of lost load (MWh) is penalised at the value of lost load, which is set to \$10,000/MWh. Publicly available demand data from the system operator for National Grid [38] and wind generation data from Whitelee wind farm [39] are used to create forecasts for 806 unique episodes, with 20 episodes held back for testing. To create power systems of different sizes (we study problems of 10–30 generators in this paper), the generators can be duplicated, an approach which has been widely adopted [14,17,40]. In this case, the demand and wind forecasts are scaled proportionally to the capacity of the generation mix.

3.2. MDP and search tree formulations

The UC problem represented in the power system environment is formalised as an episodic MDP, suitable for RL methods [4]. At each timestep, the agent receives an observation o_t consisting of the following components: (1) current generator up/down times $\boldsymbol{u}_t$; (2) demand forecast $\boldsymbol{d}_t$; (3) wind forecast $\boldsymbol{w}_t$. Demand and wind forecast errors x_t and y_t, generated by ARMA processes as described in Section 3.1, are included in the state s_t but unobserved by the agent. An action $a_t \in \{0,1\}^N$ is chosen by the agent, determining the on/off status for each of N generators (subject to generator constraints) at the next timestep. The environment processes a_t by evaluating the transition function $F(s_{t+1}, s_t, a_t)$, updating the generator up/down times, sampling forecast errors, and solving the ED problem as described in Section 3.1. The reward r_t is the negative operating cost (sum of fuel costs, startup costs and lost load costs).

The MDP can be represented as a search tree, where each node represents an observation o_t and each edge is an action a_t. Each edge has an associated cost, which is the *expected* cost of taking action a_t given observation o_t. The expected cost is estimated by calculating the mean operating cost over $N_s = 100$ scenarios of demand and wind generation, sampled from the simulation environment. Solving the UC problem amounts to finding the lowest cost path through the search tree from an initial node at $t = 1$ to any node at $t = 48$. The branching factor of the search tree is up to 2^N for N generators, intractably large for conventional tree search methods and realistic problem sizes. Using *guided expansion*, the branching factor can be intelligently reduced based on a policy trained with RL [7]. In the following section, this methodology is applied in two novel guided tree search algorithms.

[1] https://github.com/pwdemars/rl4uc.

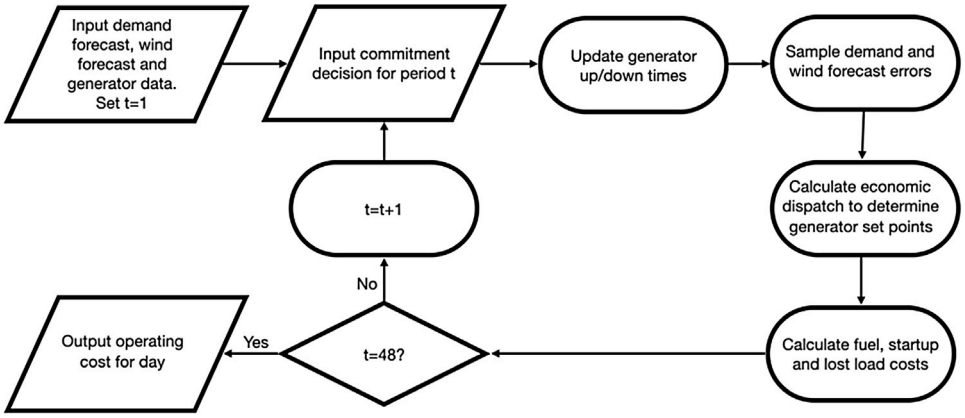

Fig. 1. Flowchart of the simulation environment. The agent inputs forecasts and generator data, and unit commitment decisions at each timestep of a 48-period day. The environment samples demand and wind scenarios and simulates dispatch by solving the economic dispatch problem. The environment outputs total operating costs at the end of the day.

4. Methodology

The guided tree search algorithm Guided UCS described in [7] uses an RL-trained policy to reduce the branching factor of a search tree. The run time of Guided UCS is highly variable across problem instances and grows exponentially with the depth parameter H. As a result, the search depth was limited to $H = 4$, or 2 hours [7]. To improve the search efficiency and reduce the run time variability, we extend this methodology to *informed* and *anytime* tree search methods with two new algorithms: Guided A* and Guided IDA*. Informed methods offer efficiency improvements by employing domain-specific knowledge, while anytime methods offer practical benefits in time-constrained contexts by being interruptible, allowing computational resources to be fully exploited for a given time budget. Before describing Guided A* and Guided IDA* search algorithms, we briefly describe guided expansion, the key innovation of guided tree search enabling the integration of RL and tree search.

4.1. Guided expansion

Guided tree search [7] uses an RL-trained policy to reduce the branching factor of a search tree. A schematic of guided tree search is shown in Fig. 2. The policy $\pi(a|s) = \Pr(A_t = a|S_t = s)$ maps states to a probability distribution over actions. The mechanism by which the branching factor is reduced is *guided expansion*. Using a pre-determined branching threshold ρ controlling the breadth of the search tree, guided expansion is used to reduce the full action space $A(s)$ to a subset $A_\pi(s)$:

$$A_\pi(s) = \{a \in A(s) | \pi(a|s) \geq \rho\} \tag{1}$$

In addition, the 'do nothing' action keeping all generator commitments the same (no startups or shutdowns) is always added to the search tree. The branching factor of the search tree can be controlled by the parameter ρ and is limited to $|A_\pi(s)| \leq \frac{1}{\rho} + 1$. In this paper we use policies from [7], trained using the policy gradient algorithm PPO.

Given a policy $\pi(a|s)$, guided expansion can be combined in a modular fashion with any tree search algorithm, creating a broader class of guided tree search algorithms. Guided expansion was applied to UCS in [7], a simple, heuristic-free algorithm that can be applied to search trees with non-uniform costs [41]. However, exploiting domain knowledge through *informed* search algorithms can significantly improve the efficiency of tree search [41]. In the following sections we apply guided expansion to A* search [10] (Guided A* search) and iterative-deepening A* search (Guided IDA* search) [11]. Guided A* is an informed search algorithm, while Guided IDA* is both informed and anytime.

4.2. Guided A* search

First, we present Guided A* search, in which guided expansion is applied to A* search [10]. Like Guided UCS, guided expansion (Eq. (1)) is used in Guided A* to reduce the branching factor of the search tree using a policy trained with RL. A* search is then used to find the lowest cost path through the reduced search tree. A* is a well-known informed search algorithm that is similar to uniform-cost search (UCS) [10]. Unlike UCS, a problem-specific heuristic function $h(n)$ estimating the optimal path cost from n to a goal node (cost-to-go) is used in A* search to determine the order in which nodes are visited and expanded. As in UCS, unexpanded nodes are stored in a priority queue data structure. However, whereas in UCS, nodes are ordered by their path costs $g(n)$, A* orders nodes by:

$$f(n) = g(n) + h(n) \tag{2}$$

Nodes with high estimated cost-to-go $h(n)$ are therefore less favourable and are relegated in the priority queue as compared with UCS. A* search is optimal if the heuristic $h(n)$ is *admissible*, meaning it underestimates the optimal cost-to-go $h^*(n)$ [42]:

$$h(n) \leq h^*(n) \tag{3}$$

Using an admissible heuristic, A* search is at least as efficient as UCS as measured by the number of node evaluations required to reach an optimal solution [10]. While admissibility is necessary to guarantee the optimality of A* search, in some contexts an inadmissible heuristic may still be effective in practice if optimal solutions are not required [43]. There are no established heuristics for the UC problem; in Section 4.4 we propose a near-admissible heuristic algorithm for UC based on priority list UC methods.

Even after applying guided expansion to reduce the branching factor, solving the entire search tree from the root node up to a depth of 48 periods is intractably expensive due to exponential run time complexity in the number of decision periods. As a result, we adopt a *real-time* strategy [44] as employed in [7]. In the real-time case, Guided A* is used to solve 48 sub-problems in a sequential manner. For each sub-problem, Guided A* search is used to find the least cost path from the root node corresponding to state s up to a search depth H. After the sub-problem has been solved, the first action a_t in the solution path is taken, and a new sub-problem is solved, this time rooted at s_{t+1}, the node following branch a_t from s_t. Combining the real-time strategy and guided expansion, Guided A* search has a run time complexity of $\mathcal{O}(T(\frac{1}{\rho})^H)$. Like Guided UCS, run time is sensitive to the breadth and depth parameters ρ and H which can adjusted to trade-off run time and solution quality but typically results in high run time variability across problem instances. Guided IDA*, described in the next section, improves on Guided A* by replacing the depth parameter H with a time budget b, making the algorithm anytime.

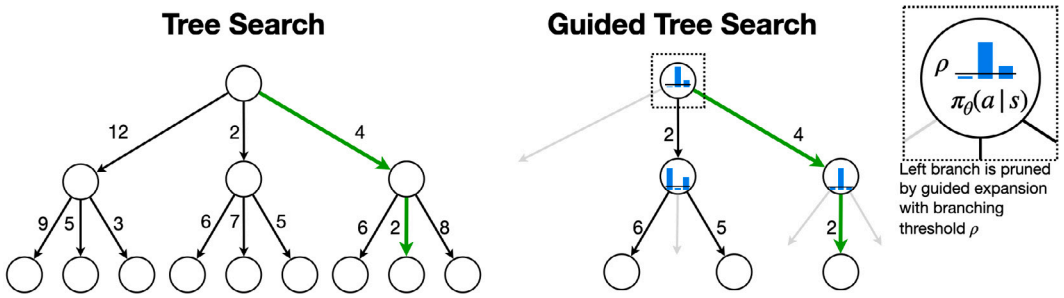

Fig. 2. Comparison of conventional tree search and guided tree search. Nodes represents states and edges represent actions. Numeric values indicate the edge costs (negative reward in the MDP formulation). Using guided expansion (Eq. (1)), a policy $\pi(a|s)$ is used to intelligently remove low probability actions from the search tree. The least cost path through the reduced search tree can be found using conventional search methods such as uniform-cost search (UCS) [8] or A* search [10].

Algorithm 1 Anytime IDA* search algorithm for the UC problem from initial state r. A* search is run with progressively increasing search horizon H until the time budget b is spent.

```
function IDAStar(r, b)
    H ← 1
    repeat
        solution ← AStar(r, H)
        H ← H + 1
    until time budget b is spent
    return solution
end function
```

4.3. Guided IDA* search

In this section we describe an *anytime* algorithm, based on iterative-deepening A* (IDA*) search. Anytime (or interruptible) algorithms can be terminated at any point and return a solution [41]. In Section 5.2, we show that Guided A* and Guided UCS exhibit high run time variability across problem instances, with over an order of magnitude separating the shortest and longest episode run times. Run time is highly unpredictable and depends on characteristics of the episode (such as demand variation) and the settings of depth and breadth parameters H and ρ. This limits the value of these methods, as optimisation problems in power systems are typically time-constrained; UC solutions are generally required within minutes [2].

Iterative deepening [11], is a general strategy that has been applied to a wide range of tree search algorithms and can be used to create anytime algorithms. The principle of iterative deepening is to gradually increase the search depth until a stopping criterion is met, such as a run time limit. A sub-optimal first action is found immediately by searching to a depth of $H = 1$. Thereafter, H is increased at each iteration and the search is conducted again. In general, solution quality improves the longer the algorithm is run due to the greater search depth. We apply iterative-deepening to the A* search algorithm described in Section 4.2; pseudocode is shown in Algorithm 1. Our implementation of IDA* replaces the fixed depth parameter H in A* with a time budget parameter b. A* search is used to iteratively solve the sub-problem rooted at r, incrementing H at each iteration. When the time budget b has elapsed, the last solution is returned. In the context of UC, the time budget b can be determined by market constraints, such as the time to market settlement. Using an anytime algorithm like Guided IDA* ensures that computational resources are fully exploited within a time constraint.

Both Guided A* and Guided IDA* require a problem-specific heuristic $h(n)$ to be applied to the UC problem. In the next section we present a heuristics based on a PL algorithm for estimating the optimal cost-to-go $h^*(n)$ in Guided A* and Guided IDA*.

4.4. Priority list heuristic

The key contribution of this work is the use of advanced search methods Guided A* and Guided IDA* which incorporate a heuristic to improve search efficiency. The heuristic $h(n)$ is problem-specific and estimates $h^*(n)$, the cost of the optimal path from node n to a goal node. The heuristic can be used to identify promising branches or prune sub-optimal ones. There is no all-purpose approach to designing effective heuristics for a particular problem domain. Some widely-studied problems have well-established heuristics. In path-finding problems, where A* search is widely applied [45,46], a common admissible heuristic is the straight-line distance from the root node to the destination node. The straight-line distance is used to calculate $h(n)$ in an application of A* search for electricity network planning in [47]. Alternatively, expert pattern databases may be used in some problems, such as the Rubik's cube puzzle [48]. Supervised learning has also been used to learn $h(n)$ for route planning problems [49]. The choice of heuristic has a significant impact on the efficiency of informed search algorithms [41]. To the best of our knowledge, no existing literature has applied informed search methods to solve the UC problem, and a new heuristic approach is required in order to apply Guided A* and Guided IDA* algorithms.

The heuristic presented in this paper is based on priority list (PL) methods, which were employed for practical UC applications in early power systems [50–52] and have also been the subject of more recent research due to their fast run times [12,53]. Improvements in MILP have made PL methods largely obsolete for practical UC problems due to their lack of optimality guarantees and reliance on complex rules to fix constraints. However, as a method for operating cost estimation where adherence to generator constraints is not strictly required, PL algorithms are a useful framework due to their low computational cost.

4.4.1. Heuristic algorithm

The heuristic proposed for the UC problem is based on a PL ordering of generators by their *minimum marginal fuel cost* (MMFC). The MMFC, denoted q_i is the marginal fuel cost ($/MWh) of generator i when operating at maximum rated capacity, $p_i^{\max}$:

$$q_i = \frac{C^f(p_i^{\max})}{p_i^{\max} t_p} \tag{4}$$

where $C_i^f(p)$ is the fuel cost function for generator i (represented as a quadratic curve in our case) evaluated for power p and t_p is the settlement period length (in hours).

To estimate optimal cost-to-go up to H timesteps ahead, the PL heuristic commits generation in PL order (i.e. in increasing order of q_i) until forecast demand is met, with no reserve constraints. Generators which are unavailable at the first scheduling period due to minimum up/down time constraints must remain online/offline until these constraints are satisfied, but thereafter these constraints are ignored. Partially relaxing inter-temporal constraints reduces the complexity of the problem and reduces the total run time of the PL algorithm. Using

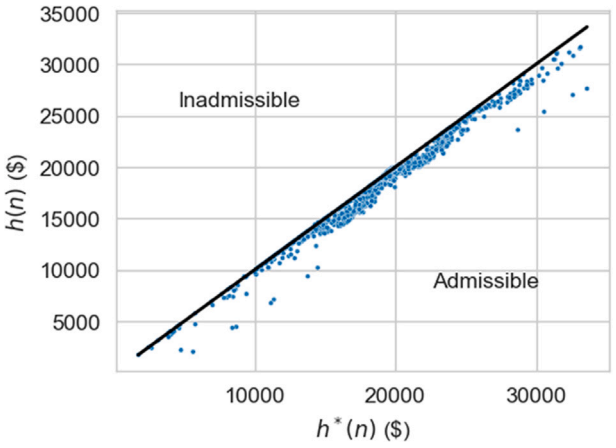

Fig. 3. Predicted cost-to-go $h(n)$ calculated with the PL heuristic and optimal cost-to-go $h^*(n)$ calculated using UCS with $H = 2$ for nodes sampled from 20 UC problem instances. The black line shows $h(n) = h^*(n)$: points below the line are admissible estimates where $h(n) \leq h^*(n)$.

this simple algorithm, a commitment schedule for the next H timesteps can be rapidly produced. To estimate the fuel costs of this schedule, the ED problem is solved with the lambda-iteration method [1] for each period based on the forecasts for demand and wind generation. The heuristic $h(n)$ is equal to the sum of fuel costs over H periods. By omitting startup costs, lost load costs, reserve constraints and some inter-temporal constraints, the PL cost is an optimistic estimate of future costs and hence more likely to produce admissible estimates where $h(n) \leq h^*(n)$. The proposed heuristic is evaluated experimentally in the following section in terms of admissibility and accuracy.

4.4.2. Heuristic accuracy and admissibility

We estimated the accuracy and admissibility of the PL heuristic by comparing estimates $h(n)$ with the optimal cost-to-go $h^*(n)$ for nodes sampled from 20 problem instances of a 5 generator power system. For each node n, we used the search algorithm UCS [8] to determine $h^*(n)$, optimally solving the least cost path problem from node up to a horizon of $H = 2$. We then used the PL heuristic to calculate $h(n)$ with the same search horizon of 2 timesteps. Evaluating the heuristic on larger power systems or with longer time horizons was not possible due to the $\mathcal{O}(2^{NH})$ run time complexity of UCS for N generators and search depth H, making the exact calculation of $h^*(n)$ intractable.

The results are plotted in Fig. 3, comparing the heuristic cost-to-go $h(n)$ and optimal $h^*(n)$. The mean absolute percentage error of $h(n) - h^*(n)$ is 3.78%. 98% of estimates $h(n)$ are admissible, indicated by points in the region below the line $h(n) = h^*(n)$. While the PL heuristic is not strictly admissible, it generally produces accurate estimates of $h^*(n)$ and in practice the operating cost differences between Guided A* and Guided UCS are negligible, as shown in Section 5.2. In addition, we show that large run time reductions are achieved by employing the PL heuristic in Guided A*.

5. Results

In this section we use Guided A* and Guided IDA* to solve UC problem instances for power systems of 10–30 generators. UC solutions are compared in terms of run time, total operating costs, and loss of load probability (LOLP). While lost load events are penalised as part of the operating cost at the value of lost load, as described Section 3.1, LOLP is an important metric to evaluate in isolation as a measure of security of supply, which may be valued over generator operating costs by system operators. We compare performance with Guided UCS from prior research, which was shown to outperform conventional MILP methods by 0.3–0.9% in terms of operating costs [7].

Table 2
Comparison of mean run time and operating cost using Guided A* search with the PL heuristic and Guided UCS [7]. Guided A* achieves significant run time reductions, with only very small changes in operating costs.

Generators	Time (% of UCS)	Cost (% of UCS)
10	6.41	100.00
20	35.62	100.03
30	17.74	100.08

5.1. Experimental setup

To allow for direct comparison between the tree search methods developed in this paper with prior work using guided tree search, our experiments use the previously trained policies [7]. To solve the 20 held-out test problems with Guided A*, we set the branching factor $\rho = 0.05$ and the search depth $H = 4$, the same parameters used for Guided UCS in [7]. For Guided IDA*, we set $\rho = 0.05$ and varied the time budget $b \in \{1, 2, 5, 10, 30, 60\}$ seconds per period to investigate the impact of run time on operating costs. Each problem instance was solved using a single Intel Xeon Gold 6140 2.30 GHz core.

To estimate the expected operating cost of solutions to the 20 UC problem instances, we applied the following Monte Carlo approach [3, 23]:

1. Calculate the UC schedule using solution method (e.g. Guided A*, Guided IDA*) based on forecasts for demand and wind.
2. Use the power system environment to calculate operating costs for $N_{\text{sim}} = 1000$ scenarios of demand and wind.

Step 2 involves calculating the real-time dispatch costs under multiple realisations of uncertainty by repeated evaluation of the UC solution using the environment described in Section 3.1. At each iteration, different demand and wind forecast errors are sampled. This method returns a distribution of operating costs over the N_{sim} scenarios, enabling solutions to be compared in terms of expected operating costs.

5.2. Heuristic evaluation

To evaluate the impact of the PL heuristic on search efficiency, we compared Guided A* search (informed) with Guided UCS (uninformed) used in previous research [7]. Total operating costs over the 20 problem instances as well as run times are compared in Table 2 for each power system size. Guided A* achieves significant run time reductions of between 64%–94%. There are small differences in operating costs of up to 0.08% between Guided UCS and Guided A*, deriving from inadmissible estimates using the PL heuristic. Overall, the deterioration in solution quality is negligible in comparison to the large run time reduction.

Fig. 4 shows the distribution of run times across problem instances for Guided A* search and Guided UCS. Both methods exhibit run time variations of roughly an order of magnitude; the 20 generator case exhibits the most extreme variation, with a factor of 36 separating the shortest and longest episode run times. The significant run time variability is caused by the differing complexity of problem instances, as illustrated in Fig. 5 comparing the 10-generator problems solved by Guided A* with shortest and longest run times. The solution for Sunday 25th June 2017 is shown in the left-hand plot, a simple problem instance with little demand variation and low wind penetration, requiring only 3 changes in commitment. The right-hand plot shows the solution for Monday 21st November 2016, a more complex problem instance with greater wind penetration and higher and more variable demand. Increasing wind penetration at the end of the day causes a sharp decline in net demand that necessitates the decommitment of several generators to avoid encountering the generation floor — the sum of minimum operating levels of online generators. Using guided tree search methods,

Table 3
Comparison of Guided IDA* (b = 30s), Guided A* (H = 4) and Guided UCS (H = 4) [7] for 10, 20 and 30 generator problems. Mean cost indicates the mean total operating costs over N_{sim} = 1000 realisations of demand and wind generation. $\hat{t}$ represents the mean run time across the 20 problem instances. The variability of run time is represented by t_{max}/t_{min}, the ratio of maximum to minimum run time. Loss of load probability is the proportion of periods during which lost load was experienced during 1000 simulations of the 20 problem instances.

Num. gens	Method	Heuristic	Cost ($M)	$\hat{t}$ (s)	t_{min} (s)	t_{max} (s)	t_{max}/t_{min}	LOLP (%)
10	IDA*	PL	9.33	1086.3	892.0	1294.1	1.5	0.12
	A*	PL	9.37	51.8	5.7	195.8	34.4	0.13
	UCS	None	9.37	807.3	76.9	1992.1	25.9	0.13
20	IDA*	PL	18.67	1099.6	797.0	1267.6	1.6	0.12
	A*	PL	18.74	41.8	7.7	159.3	20.7	0.11
	UCS	None	18.73	117.3	10.4	374.5	36.0	0.11
30	IDA*	PL	28.14	1210.0	986.0	1359.7	1.4	0.11
	A*	PL	28.43	69.4	14.5	164.4	11.3	0.14
	UCS	None	28.41	391.3	129.4	976.8	7.5	0.14

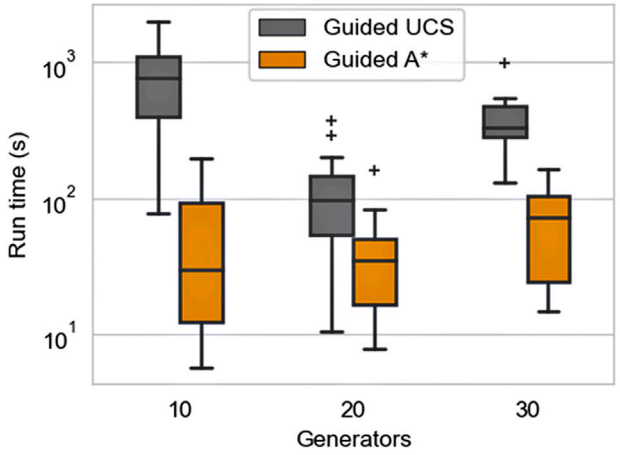

Fig. 4. Comparison of run time (log-axis) between Guided A* search and Guided UCS. The PL heuristic achieves mean run time reductions of between 64%–94%. In both cases, there is large variability in run time between UC problem instances.

more complex problems typically result in broader search as the policy $\pi(a|s)$ recommends more actions to explore, resulting in longer run times.

The run time of guided tree search algorithms Guided A* and Guided UCS is difficult to predict and is sensitive to the complexity of the problem instance as well as the search depth H and branching threshold ρ. As discussed in Section 4.3, this motivated the development of the anytime algorithm Guided IDA*. In the following section, we apply this algorithm to solve the same problem instances and compare its performance with Guided A* and Guided UCS.

5.3. Comparison of tree search methods

Table 3 compares the performance of Guided IDA*, Guided A* and Guided UCS [7] for 20 problem instances with 10, 20 and 30 generator systems. For Guided IDA*, results are shown for b = 30s (maximum 1440s per episode), comparable to the maximum run time of Guided UCS. Guided IDA* achieves lower operating costs than both other guided tree search methods, with a maximum run time that is similar to that of Guided UCS. The LOLP remains similar across all three guided tree search methods. The run time variability is measured as the ratio of maximum to minimum run time t_{max}/t_{min}. Using non-anytime methods Guided UCS and Guided A*, run time variability is between 7.5–34; using Guided IDA* search, variability is reduced to between 1.4–1.6.

Improvements in operating costs can be attributed to greater average search depths using Guided IDA* compared with Guided UCS and Guided A*, where search depth was fixed at H = 4. Fig. 6 shows the median search depth H using Guided IDA* with varying time budget b. Even with the lowest time budget of b = 1 second per period, the median search depth is $H \geq 5$ and at larger time budgets is significantly higher than H = 4 used for UCS and A*. Median search depth increases logarithmically with respect to the budget b, increasing by one for approximately each doubling of b. This means that relatively deep search is achieved on average, even for small time budgets.

Fig. 7 shows the impact of the time budget b on operating costs for Guided IDA*, with comparison to Guided UCS. While costs were generally found to decrease with increasing time budget, there is a notably non-uniform decrease for the 30 generator problem for $b \leq$ 10s. Guided IDA* outperforms Guided UCS for budgets $b \geq$ 10s for all three problem sizes. The largest savings compared with UCS were achieved in the 30 generator case, where costs were 1.1% lower with b = 60.

The lower run time variability of Guided IDA* is a practical advantage over Guided UCS and Guided A*, and enables more reliable performance characteristics by maximising use of computational resources. Our results show that for comparable time budgets, Guided IDA* achieves significant cost savings as compared with Guided UCS.

6. Discussion

This paper built on the methodology of previous research combining model-free RL with model-based planning [7], addressing limitations of run time variability and limited search depth. Our results showed that the choice of tree search algorithm (UCS, A* or IDA*) is an important design decision in this broader class of guided tree search methods, and has a significant impact on solution quality and run times. Using informed and anytime search methods yielded substantial performance improvements and practical benefits as compared with the uninformed, non-anytime algorithm Guided UCS.

A key contribution of this research is a heuristic for the UC problem, which achieved substantial run time reductions of up to an order of magnitude when applied in Guided A* search. The development of the heuristic demanded domain knowledge of power systems to balance accuracy with run time. While problem-agnostic blackbox methods have been used in other contexts to develop heuristics for informed search [49], problem-specific deterministic methods can incorporate human knowledge of the problem and may be more reliable in practice. Complementing RL expertise with problem-specific knowledge is an effective and valuable means of improving solution quality in applied contexts which can help accelerate the adoption of machine learning methods for practical benefit.

The complexity of UC problem instances varies significantly depending on the characteristics of demand and renewables generation forecasts which can result in substantial variation in computational expense for RL methods. Fig. 5 contrasted the problem complexities of a weekend summer day and a winter weekday, whose run times using Guided A* were separated by a factor of 34. A successful solution method must be one that is capable of solving diverse problem instances without significant loss of solution quality or increase in run time. In the anytime algorithm Guided IDA*, a fixed time budget ensures that run times do not become impractically large in complex problem

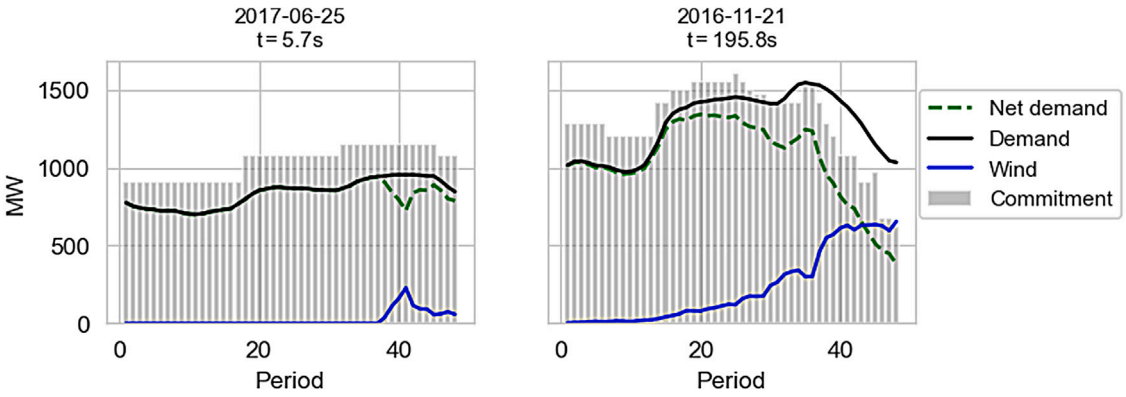

Fig. 5. Comparison of two 10 generator UC problem instances solved with Guided A* search. The left-hand problem Sunday 25th June 2017 was solved in the shortest time ($t = 5.7$ s) and is characterised by low wind penetration and flat demand. The right-hand plot shows the solution for Monday 21st November 2016 ($t = 195.8$ s) and is characterised by more variable demand and increasing wind penetration that coincides with falling demand after the evening peak. This varying complexity across problem instances is the cause of large run time variability using Guided A* search.

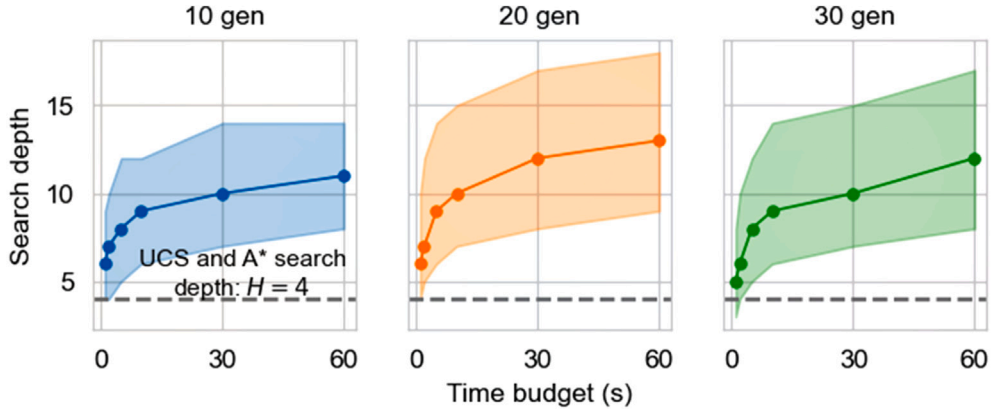

Fig. 6. Search depth of Guided IDA* for 10, 20 and 30 generator problem instances. Solid line and points show the median search depth; shaded area indicates inter-quartile range. Dotted line shows $H = 4$, the search depth used in Guided UCS [7] and Guided A* search. For all time budgets, the average search depth of Guided IDA* is significantly greater than other guided tree search methods.

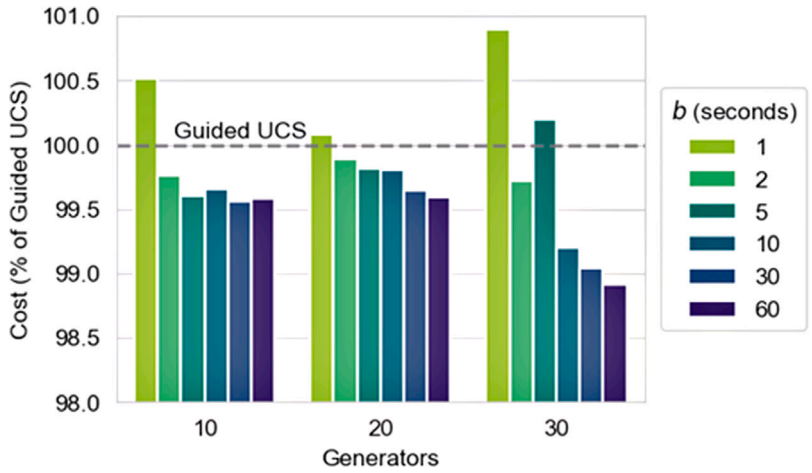

Fig. 7. Cost saving of Guided IDA* with PL heuristic compared to Guided UCS. Operating costs generally decrease with increasing time budget. The largest improvements are found in 30 generator case, where IDA* is 1.1% cheaper than Guided UCS when $b = 60$ s.

instances, compensating with an adaptive search depth. In practice, Guided IDA* reaches greater search depths on average, as shown in Fig. 6, resulting in lower operating costs for similar computational budgets. By contrast, the high sensitivity of Guided UCS and Guided A* run times to the fixed search depth H means that increasing the depth of search risks an explosion in computational expense for some problem instances. The anytime property of Guided IDA* is a significant practical advantage over UCS and A* for the UC problem, reducing run time variability and allowing for schedules to be reliably produced in time-constrained contexts. The time budget of Guided IDA* can be set

using knowledge of market constraints, such as the time to gate closure when bids and offers must be submitted.

Increasing the time budget in Guided IDA* generally resulted in operating cost reductions as shown in Fig. 7. In all three problems, Guided IDA* with a budget of $b \geq 10$ seconds per period outperformed Guided UCS. Our results did not find a deterioration in performance with increasing number of generators and the greatest operating cost reductions relative to Guided UCS were found for the largest problem instances with 30 generators. The potential for scaling to power systems with more generators is therefore promising, although challenges may be encountered in policy training as the number of actions meeting the branching threshold ρ becomes increasingly sensitive to the policy entropy. Further work on entropy regularisation for guided tree search may therefore be required for successful applications to larger power systems. Furthermore, Fig. 7 showed that for the 30 generator system, performance was more variable for lower budgets of $b \leq 5$s, indicating that greater time budgets may be required to ensure stable performance for larger systems.

7. Conclusion

Compared with previous research combining model-based and model-free RL [7], the two novel algorithms developed in this paper, Guided A* and Guided IDA*, are more effective solution methods for the UC problem and represent a significant advance in the field of RL for UC. Using Guided A* search and a novel heuristic function based on priority list solution methods [12], run times are reduced by up to 94% as compared with Guided UCS, with negligible (< 0.1%) impact on operating costs. These results demonstrate the value of domain expertise in designing solution methods for UC and other real-world problems. While a large proportion of RL literature has studied games-playing domains, research in real-world contexts has progressed more slowly [54]. Our results show that combining domain expertise with state of the art RL can improve solution methods for specific applications, accelerating the adoption of RL methods for practical benefit.

The UC problem is typically highly time-constrained, and must usually be solved within minutes [2]. The variable and unpredictable run times of fixed-depth tree search methods such as UCS and A* across problem instances of varying complexity therefore pose practical problems for UC. We developed an anytime algorithm, Guided IDA*, to mitigate run time variability, constraining the run time to a fixed computational budget. Guided IDA* achieved operating cost reductions of up to 1% for similar computational budgets as compared with Guided UCS, similar to the cost savings shown by stochastic optimisation over deterministic methods using MILP [3,23,25]. Anytime methods were shown to be particularly well-suited to the UC problem, enabling more reliable generation of high-quality solutions as compared with fixed-depth tree search.

Previous research showed that the exponential time complexity of tree search algorithms can be overcome by employing an RL policy to reduce the branching factor of the search tree [7]. This paper has shown that this methodology can be further improved for UC by modifying the algorithm to exploit properties of the problem in Guided A* and Guided IDA*. While conventional tree search methods are impractical for UC applications, this paper has shown that problem-specific modifications using RL and advanced search methods can enable tree search to be successfully applied, producing high quality solutions.

RL offers several advantages over mathematical optimisation techniques for UC including principled accounting of uncertainty, absence of heuristic reserve requirements and offline training which reduces the computational burden at decision time. However, in order for RL to become a viable method for practical use, further work is required to verify its superiority over existing methods and build trust among power system operators and generating companies. As topics for further research, studies of problem instances with transmission network constraints, generator outages and profit-based agents would show the generality of RL across diverse real world contexts.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the use of UCL's Myriad High Performance Computing cluster for this research. This research was supported by an Engineering and Physical Sciences Research Council research studentship (grant number: EP/R512400/1).

References

[1] Wood AJ, Wollenberg BF, Sheblé GB. Power generation, operation, and control. John Wiley & Sons; 2013.
[2] Knueven B, Ostrowski J, Watson J-P. On mixed-integer programming formulations for the unit commitment problem. INFORMS J Comput 2020;32(4):857–76.
[3] Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T. Adaptive robust optimization for the security constrained unit commitment problem. IEEE Trans Power Syst 2012;28(1):52–63.
[4] Sutton RS, Barto AG, et al. Introduction to reinforcement learning, Vol. 135. MIT press Cambridge; 1998.
[5] Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, et al. Mastering atari, go, chess and shogi by planning with a learned model. Nature 2020;588(7839):604–9.
[6] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science 2018;362(6419):1140–4.
[7] de Mars P, O'Sullivan A. Applying reinforcement learning and tree search to the unit commitment problem. Appl Energy 2021;302:117519.
[8] Dijkstra EW, et al. A note on two problems in connexion with graphs. Numer Math 1959;1(1):269–71.
[9] de Mars P, O'Sullivan A, Keppo I. Estimating the impact of variable renewable energy on base-load cycling in the GB power system. Energy 2020;195:117041.
[10] Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 1968;4(2):100–7.
[11] Korf RE. Depth-first iterative-deepening: An optimal admissible tree search. Artificial Intelligence 1985;27(1):97–109.
[12] Senjyu T, Shimabukuro K, Uezato K, Funabashi T. A fast technique for unit commitment problem by extended priority list. IEEE Trans Power Syst 2003;18(2):882–8.
[13] Håberg M. Fundamentals and recent developments in stochastic unit commitment. Int J Electr Power Energy Syst 2019;109:38–48.
[14] Carrión M, Arroyo JM. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans Power Syst 2006;21(3):1371–8.
[15] Muckstadt JA, Koenig SA. An application of Lagrangian relaxation to scheduling in power-generation systems. Oper Res 1977;25(3):387–403.
[16] Fu Y, Shahidehpour M, Li Z. Security-constrained unit commitment with AC constraints. IEEE Trans Power Syst 2005;20(2):1001–13.
[17] Kazarlis SA, Bakirtzis A, Petridis V. A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 1996;11(1):83–92.
[18] Chakraborty S, Ito T, Senjyu T, Saber AY. Unit commitment strategy of thermal generators by using advanced fuzzy controlled binary particle swarm optimization algorithm. Int J Electr Power Energy Syst 2012;43(1):1072–80.
[19] Zhuang F, Galiana F. Unit commitment by simulated annealing. IEEE Trans Power Syst 1990;5(1):311–8.
[20] Nycander E, Morales-España G, Söder L. Security constrained unit commitment with continuous time-varying reserves. Electr Power Syst Res 2021;199:107276.
[21] Holttinen H, Milligan M, Kirby B, Acker T, Neimane V, Molinski T. Using standard deviation as a measure of increased operational reserve requirement for wind power. Wind Eng 2008;32(4):355–77.
[22] Hedman KW, Ferris MC, O'Neill RP, Fisher EB, Oren SS. Co-optimization of generation unit commitment and transmission switching with N-1 reliability. IEEE Trans Power Syst 2010;25(2):1052–63.
[23] Ruiz PA, Philbrick CR, Zak E, Cheung KW, Sauer PW. Uncertainty management in the unit commitment problem. IEEE Trans Power Syst 2009;24(2):642–51.
[24] Bouffard F, Galiana FD. Stochastic security for operations planning with significant wind power generation. In: 2008 IEEE power and energy society general meeting-conversion and delivery of electrical energy in the 21st century. IEEE; 2008, p. 1–11.
[25] Tuohy A, Meibom P, Denny E, O'Malley M. Unit commitment for systems with significant wind penetration. IEEE Trans Power Syst 2009;24(2):592–601.

[26] Papavasiliou A, Oren SS, Rountree B. Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration. IEEE Trans Power Syst 2014;30(3):1109–20.
[27] Jasmin E, Ahamed TI. Reinforcement learning solution for unit commitment problem through pursuit method. In: 2009 International conference on advances in computing, control, and telecommunication technologies. IEEE; 2009, p. 324–7.
[28] Jasmin E, Ahamed TI, Remani T. A function approximation approach to reinforcement learning for solving unit commitment problem with photo voltaic sources. In: 2016 IEEE international conference on power electronics, drives and energy systems. IEEE; 2016, p. 1–6.
[29] Li F, Qin J, Zheng WX. Distributed *Q*-learning-based online optimization algorithm for unit commitment and dispatch in smart grid. IEEE Trans Cybern 2019;50(9):4146–56.
[30] Navin NK, Sharma R. A fuzzy reinforcement learning approach to thermal unit commitment problem. Neural Comput Appl 2019;31(3):737–50.
[31] Dalal G, Mannor S. Reinforcement learning for the unit commitment problem. In: 2015 IEEE eindhoven powertech. IEEE; 2015, p. 1–6.
[32] Qin J, Yu N, Gao Y. Solving unit commitment problems with multi-step deep reinforcement learning. In: 2021 IEEE international conference on communications, control, and computing technologies for smart grids. IEEE; 2021, p. 140–5.
[33] Glavic M, Fonteneau R, Ernst D. Reinforcement learning for electric power system decision and control: Past considerations and perspectives. IFAC-PapersOnLine 2017;50(1):6918–27.
[34] Rolnick D, Donti PL, Kaack LH, Kochanski K, Lacoste A, Sankaran K, et al. Tackling climate change with machine learning. 2019, arXiv preprint arXiv:1906.05433.
[35] Perera A, Kamalaruban P. Applications of reinforcement learning in energy systems. Renew Sustain Energy Rev 2021;137:110618.
[36] Dalal G, Gilboa E, Mannor S. Hierarchical decision making in electricity grid management. In: International conference on machine learning. PMLR; 2016, p. 2197–206.
[37] Dulac-Arnold G, Mankowitz D, Hester T. Challenges of real-world reinforcement learning. 2019, arXiv preprint arXiv:1904.12901.
[38] National Grid Demand Data, https://www.nationalgrideso.com/data-explorer.
[39] Balancing Mechanism Reporting Service, https://www.bmreports.com.
[40] Ostrowski J, Anjos MF, Vannelli A. Tight mixed integer linear programming formulations for the unit commitment problem. IEEE Trans Power Syst 2011;27(1):39–46.
[41] Russell S, Norvig P. Artificial intelligence: A modern approach. 3rd ed.. USA: Prentice Hall Press; 2009.
[42] Dechter R, Pearl J. Generalized best-first search strategies and the optimality of A. J ACM 1985;32(3):505–36.
[43] Ernandes M, Gori M. Likely-admissible and sub-symbolic heuristics. In: ECAI, Vol. 16. Citeseer; 2004, p. 613.
[44] Korf RE. Real-time heuristic search. Artificial Intelligence 1990;42(2–3):189–211.
[45] Golden BL, Ball M. Shortest paths with euclidean distances: An explanatory model. Networks 1978;8(4):297–314.
[46] Sedgewick R, Vitter JS. Shortest paths in euclidean graphs. Algorithmica 1986;1(1–4):31–48.
[47] Li JC, Zimmerle D, Young PM. Effective rural electrification via optimal network: Optimal path-finding in highly anisotropic search space using multiplier-accelerated A* algorithm. Energy AI 2022;7:100119.
[48] Korf RE, Reid M, Edelkamp S. Time complexity of iterative-deepening-A*. Artificial Intelligence 2001;129(1–2):199–218.
[49] Wang J, Wu N, Zhao WX, Peng F, Lin X. Empowering a* search algorithms with neural networks for personalized route recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, p. 539–47.
[50] Kerr R, Scheidt J, Fontanna A, Wiley J. Unit commitment. IEEE Trans Power Appar Syst 1966;(5):417–21.
[51] Baldwin C, Dale K, Dittrich R. A study of the economic shutdown of generating units in daily dispatch. Trans Am Inst Electr Eng. Part III: Power Apparat Syst 1959;78(4):1272–82.
[52] Johnson R, Happ H, Wright W. Large scale hydro-thermal unit commitment-method and results. IEEE Trans Power Appar Syst 1971;(3):1373–84.
[53] Quan R, Jian J, Yang L. An improved priority list and neighborhood search method for unit commitment. Int J Electr Power Energy Syst 2015;67:278–85.
[54] Dulac-Arnold G, Evans R, van Hasselt H, Sunehag P, Lillicrap T, Hunt J, et al. Deep reinforcement learning in large discrete action spaces. 2015, arXiv preprint arXiv:1512.07679.