Georgievski *et al. Energy Informatics* 2023, **6**(Suppl 1):18
https://doi.org/10.1186/s42162-023-00289-w

Energy Informatics

RESEARCH

Open Access

AI temporal planning for energy smart buildings

Ilche Georgievski[1*], Muhammad Zamik Shahid[2] and Marco Aiello[1]

From The 12th DACH+ Conference on Energy Informatics 2023
Vienna, Austria. 4-6 October 2023. https://www.energy-informatics2023.org/

*Correspondence:
ilche.georgievski@iaas.uni-stuttgart.de

[1] Service Computing Department, IAAS, University of Stuttgart, Universitaetsstrasse 38, 70569 Stuttgart, Germany
[2] M.Sc. Student in Information Technology, University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany

Abstract

Buildings are responsible for about one-third of industrialised countries' overall energy consumption and greenhouse gas emissions. As if this was not enough, recently, energy prices significantly increased and affected all economic areas. Making buildings more efficient and effective is the step needed toward cost reductions. Key enablers of cost-effectiveness are leveraging batteries, awareness of and adaptability to energy prices, and integrating powerful reasoning techniques to optimally and flexibly operate buildings. Researchers have tackled many of these aspects using a variety of approaches. Whereas a less investigated one is that of AI planning to coordinate actions and save energy in buildings. However, generating plans based on signals of energy prices and leveraging batteries is still an open research problem. To address this high-potential aspect, we engineer an AI planning system for improving the energy-cost effectiveness in buildings by coordinating the building's operation based on day-ahead prices and the use of a battery, all without sacrificing the comfort of building occupants. We propose to exploit temporal planning due to its powerful modelling and reasoning features, especially in explicitly addressing time. We evaluate the effectiveness of the system in several scenarios with varying building environmental conditions. We compare the energy cost from using our planning system to a baseline cost, where we record a reduction of 43rage in favour of our system.

Keywords: Energy cost, Temporal planning, Engineering AI planning systems, Smart buildings

Introduction

We recently witnessed a substantial increase in energy costs. At some point in 2021, electricity prices leapt to their highest level on record in Germany, more than six times compared to 2020 (IEA 2021). Similar situations were experienced in most countries worldwide. Consequently, various sectors and levels have introduced pressing measures to save energy. For example, the German federal government has issued Ordinances on Securing the Energy Supply through Rapid and Medium-Term Impact Measures (BMWk 2022). Non-residential buildings are in the spotlight because they significantly contribute to energy consumption and CO_2 emissions. The International Energy Agency reports that 30% of global final energy consumption and 27% of total energy sector emissions can be attributed to buildings, of which 8% are direct emissions, and 19% are indirect emissions from energy production for heating (IEA 2022). The German case is even worse: buildings account for about 35% and 30% of the total energy consumption and emissions, respectively (BMWi 2015). The ordinance measures specify lowering the building temperature and reducing lighting. Many organisations take their initiatives to motivate building occupants to help buildings save energy by "turning down the heating when rooms are not in use" or "regularly inspecting rooms and switching off devices and equipment that are not needed."[1]

These measures and initiatives indicate the need to address energy usage in buildings and the space for improvement over current practices. Some technical innovations, such as those based on the Internet of Things (IoT), deliver basic automation and quite naïve overall control. For example, if the room temperature as measured by a sensor exceeds a predefined value, the Heating, Ventilation, and Air Conditioning (HVAC) may start cooling the room down till it reaches a predefined threshold. While this approach is a step forward, it must also consider the room occupancy, comfort needs, and energy prices with a coordinated and anticipated operation. In other words, the actions of devices and systems embedded in buildings should be automatically selected, ordered, and executed to maximise occupant needs and minimise energy costs and energy consumption considering the conditions and properties of building environments. This is known as *building coordination problem* (Georgievski et al. 2017).

State of the art

Various forms of building coordination problems have been tackled with diverse techniques, such as model predictive control, rule-based systems, and machine learning. Artificial Intelligence (AI) planning is another technique that provides powerful reasoning capabilities and has the potential of addressing the building coordination problem. AI planning is concerned with the selection and organisation of actions needed to achieve a given objective (Ghallab et al. 2004). AI planning has been used to coordinate actions in various buildings, such as homes and office buildings. We have employed planning based on constraint satisfaction to select and organise device actions that satisfy high-level goals of home inhabitants (Kaldeli et al. 2013). We have also used Hierarchical Task Network planning to adapt a building's operation according to people's activities while consuming as little energy as possible (Georgievski et al. 2017). Bajada investigated the use of AI temporal planning for controlling the temperature in a given room (cf. thermostat domain) (Bajada 2016). In the same work, Bajada uses AI temporal planning to automate the *demand dispatch* for electricity load management with cost optimisation from an aggregator's perspective.[2] Shaikh studied the use of AI temporal planning for optimising the energy use of HVAC (Shaikh 2021). Their approach generates plans for HVAC components to obtain thermal comfort while minimising energy consumption.

[1] Rector's email messages about our university's campaign to save energy (University of Stuttgart 2023).

[2] Aggregators balance demand and supply to guarantee the power grid's stability.

Previous studies highlight the potential of AI planning for reducing costs and provide approaches to addressing specific building coordination problems. However, several vital factors still need to be considered. First, building environments are inherently temporal (an exception is (Shaikh 2021), which considers temporal aspects in the context of thermal comfort). Furthermore, operating loads considering their properties and constraints is crucial for minimising building costs. Loads are considered in (Bajada 2016), however, from an aggregator's perspective and not of buildings. Also, the possibility of having a dynamic price structure and being equipped with batteries is already available and provides even more opportunities for more cost-effective building coordination. Therefore, the question arises: How can one design and develop a planning system that considers the characteristics of building environments and energy pricing and storage possibilities to make smart buildings more cost-effective?

Contributions and organisation

The present work addresses the question by engineering an AI planning system for improving energy-cost effectiveness in smart non-residential buildings. We design and realise the planning system by following best practices for software-engineering ICT systems. We exploit AI temporal planning to generate effective and flexible plans for coordinating a building's operation considering various building loads, building environmental conditions, energy prices from the day-ahead energy market, and a battery. We evaluate the effectiveness of the planning system using scenario analysis and show that our approach can reduce the energy cost by 43% on average compared to a baseline cost. We also offer a discussion before we finalise the paper.

Designing the planning system

Designing building management systems is a complex process that requires critical understanding and careful consideration of buildings' various operational and technical aspects to make the systems efficient, effective, flexible, and robust. Thus, to design our planning system, we follow the software development life cycle for engineering AI planning systems (Georgievski 2023). We present the design-oriented phases next.

Requirements analysis

The first phase is about analysing the requirements the intended planning system needs to meet to ensure reasonable system quality. We focus on domain-oriented and functional requirements; the former capture the application domain knowledge, and the latter describe the planning system's functions and their inputs and outputs (Georgievski 2023).

Domain-oriented requirements

The application domain knowledge includes relevant buildings' objectives, operations, loads, batteries, day-ahead pricing, and other building characteristics. A building may have multiple objectives, such as improving comfort, saving energy, and minimising energy cost, which should be satisfied in strict total order (Aiello et al. 2021). That satisfaction involves managing building operations and loads according to relevant building constraints. Our focus is on improving the energy-cost effectiveness of buildings by managing the operations of loads and batteries concerning day-ahead energy prices without risking the comfort and safety of occupants. Given this objective, we select typical and representative operations: controlling thermal comfort (i.e., temperature) and visual comfort (i.e., light level), managing electrical equipment (i.e., dishwasher), and operating energy storage (i.e., battery) (Georgievski and Aiello 2022).

Loads refer to energy-consuming systems that may exist within buildings. Light loads are relatively low in power consumption and typically do not produce significant heat (e.g., printers). Heavy loads consume a lot of energy and are usually related to the operation of mechanical and electrical equipment (e.g., HVAC). Heavy loads can thus significantly impact the energy efficiency of buildings. While managing light loads is valuable, reducing the energy consumption of heavy loads is crucial for substantially improving a building's energy efficiency. Loads can be controllable or uncontrollable. The former can be directly controlled based on the current building's environmental conditions and/or energy pricing. Controllable loads can be dependent or independent. Dependent controllable loads consume energy depending on multiple factors, such as occupancy levels, light levels, building temperature, and energy prices. Managing these loads effectively requires careful consideration of such factors to decide when to actuate them. Independent controllable loads are directly and strictly dependent on current energy prices and are not influenced by the building's environmental conditions. These loads provide scheduling flexibility and can help reduce energy costs and consumption. Finally, uncontrollable loads are often exogenous to buildings and cannot be controlled based on building environment conditions or energy pricing. We do not consider this load type.

The battery stores and supplies energy. The state of charge (SOC) represents the energy stored in the battery, ranging from 0% to 100%. SOC is crucial in ensuring the battery is not overcharged or over-discharged, which can damage the battery's health and longevity. SOC changes with specific charging or discharging rates.

Day-ahead pricing is a mechanism used in electricity markets where the electricity price is set one day before the delivery of electricity based on the expected demand and supply conditions. Suppliers submit their bids for the electricity amount they can supply and at what price in hourly blocks, and the market operator determines the market-clearing price based on the total supply and demand.

Building operations and loads require dealing with *quantities*, such as temperature, light intensity, energy demand, energy cost, energy prices, and SOC. The values these quantities should take are often regulated by constraints (Georgievski and Aiello 2022). *Wellbeing constraints* express conditions necessary for maintaining comfortable and healthy building environments, *temporal operation constraints* define the earliest start time, latest finish time, and duration of operations of controllable loads, and *temporal business constraints* define the building operating hours. Furthermore, some controllable loads may have flexible operation or operation duration (e.g., charging battery duration can be decided on the fly). Finally, we must consider that although some domain elements cannot be controlled, they provide temporal background information (e.g., day-ahead markets provide hourly energy prices).

Functional requirements

The primary function of the planning system is to *solve* planning problems that express the domain-oriented requirements. This Solving functionality accepts planning problems specified in a well-defined syntax, such as the Planning Domain Definition Language (PDDL) (McDermott et al. 1998). PDDL requires a planning problem to be *modelled* in two separate parts: a planning domain and problem instance. Thus, Domain Modelling is another needed functionality typically accomplished by a domain or planning expert. For problem instances, which need to be generated on the fly whenever a relevant environmental change occurs, we need a functionality that can automatically *generate* them from data from various sources (e.g., IoT and energy market). Finally, as day-ahead pricing is a primary domain-oriented requirement in our work, we need a functionality that *gathers* actual day-ahead prices for an area of interest.

AI planning model formulation

In our work, formulating a planning model corresponds to selecting a planning type that supports meeting the domain-oriented requirements (Georgievski 2023). Given our domain's properties, classical planning is unsuitable because of its limitations (Ghallab et al. 2004). Temporal planning, on the other hand, supports reasoning about numbers and time, allowing modelling domains with numeric and temporal properties. Central in temporal planning are durative actions with preconditions, effects, and duration constraints, which can express that actions occur over a time span, preconditions may not need to hold only at the action's start, and effects hold during the entire action execution. Solutions to temporal planning problems are plans—sets of instantaneous actions and tuples of a starting time, durative action, and duration.

PDDL version 2.1 supports expressing numeric and temporal domain properties using numeric functions (also called fluents), durative actions, and duration inequalities (Fox and Long 2003). Representation of time-dependent and conditional states can be accomplished using timed initial literals (TILs) (Edelkamp and Hoffmann 2004) and timed initial fluent (TIFs) (Piacentini et al. 2015). TILs are predicates that hold at specific times in the initial state, while TIFs are numeric functions that take time as input and return a value that holds at that time. Both can be used to represent initial conditions that vary over time. An example of TILs are `(at 8 (operating-hour))` and `(at 20 (not (operating-hour))`, representing the time window of building operating hours. An example of TIFs is `(at 14 (= (current-temperature) 21.5))`, representing the temperature at 14:00 h. For more instructions on PDDL modelling, we refer to McDermott et al. (1998), Fox and Long (2003).

Planning domain design

The next lifecycle phase is about designing the planning domain and the choices made during this process. We represent each domain entity using a predicate. For example, a building's operating hour is represented by the following predicate without arguments `(operating-hour)`. Loads with Boolean variables are also represented using predicates, which are then used to check the current state and alter the loads' state. An example of such a predicate is `(turned-on ichl-dishwasher)`, which indicates that

```
(:durative-action charge-battery-low-price
 :parameters ()
 :duration (= ?duration 2)
 :condition (and (over all (= (price) 2)) (at start (speak-battery))
                 (at start (< (soc-battery) 100)) (at start (enable)))
 :effect (and (at end (increase (soc-battery)(cr-battery)))
              (at end (increase (energy-capacity) time-lapse))
              (at start (not (speak-battery)))
              (at end (speak-battery))))
```

Fig. 1 PDDL durative action for charging the battery at a low price

the dishwasher is turned on. We represent environment data, day-ahead prices, battery knowledge, and some numeric constraints using numeric functions. For example, the current temperature is encoded with the `(current-temperature)` function, day-ahead prices with `(energy-price)`, and the range of comfortable temperature with `(comfort-min)` and `comfort-max`. We represent the battery's knowledge using three numeric functions. One function keeps track of the battery's charge state; another represents the battery charging rate; and the third represents the battery's discharge rate to meet the building's energy demand. These numeric functions play a vital role in our domain design as we use them to identify relevant entities' optimal values and save costs and energy consumption during peak hours.

We model loads around the building operating hours, which we assume to be from 08:00 to 20:00. The electrical storage system is the only load not dependent on operating hours, thus, it would operate 24/7. Its corresponding actions are designed in a way that results in charging the battery at low energy prices and discharging the battery at high costs, and potentially selling the energy back to the grid at a higher price. This strategy significantly reduces costs, even during weekends. Outside operating hours, including weekends and public holidays when non-residential buildings are generally closed, all loads would be turned off except for the Controllable Dependent Light Loads, which, in our case, are represented by ordinary lights. The ordinary lights, such as hallway lights, would remain turned on but at a lower intensity for visibility and security purposes, consuming little energy.

We define a set of durative actions with a fixed duration that can be used to select an action for each load in the day-ahead plan. Due to space limits, we only explain the action for charging the battery at a low price. We operate the battery according to its state of charge, energy prices, and the building's demand. We use the battery's numeric functions to ensure that the battery is utilised efficiently to minimise energy costs. The value of the SOC function changes based on the battery's charging and discharging rate and the building's energy demand. The battery's charging rate varies based on the energy price. The battery is not charged during peak hours, with high energy costs. The battery only charges at low or nominal energy prices if the SOC is too low. The battery's discharging rate also varies based on the energy price, and it is higher when the energy cost is high as the necessary load to maintain a comfortable building environment is shifted to the battery at this time. We use this knowledge to model three durative actions for the battery: one for charging at low energy prices and two actions for discharging at high and nominal energy prices. Figure 1 illustrates the former action with the maximum charging rate.

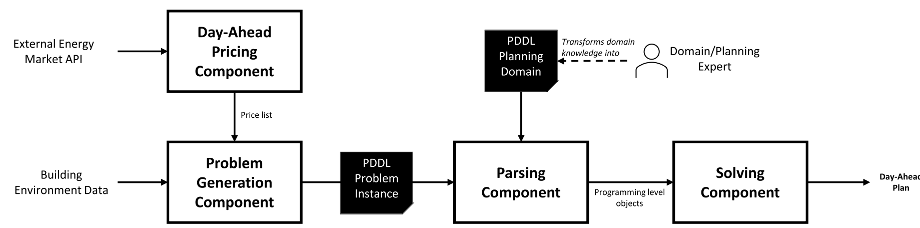


Fig. 2 Overview of the architecture design of the planning system

System architecture design

Figure 2 shows an overview of the planning system architecture. The architecture helps to ensure that all aspects of the planning system are considered, from logical components and user inputs to connections and data flow. We assume the system components are executed in the specified order and communicate synchronously.

The *Day-Ahead Pricing Component* gathers data about hourly energy prices for a day ahead in a given energy market area. It then filters and bins the data to compose data meaningful to our purpose. The binning involves dividing the prices into different ranges based on their values and thresholds. The prices are binned into three categories: high, nominal, and low. By doing so, we can easily set the price level for every hour of the day in a problem instance, enabling us to make informed decisions based on price levels. The *Problem Generation Component* accepts the energy prices and building environment data. We assume the building environment data, which can come from sensors or some repository, includes information about occupancy, natural light, current temperature, and the building's operating hours. The Problem Generation Component transforms the input data into PDDL problem specification (i.e., objects, initial state, and goal). The component puts this specification into a file and passes it to the Parsing Component. The PDDL problem instance and the PDDL planning domain model described in Planning Domain Design represent the input to the *Parsing Component*, which analyses whether planning problems conform to the PDDL rules and transforms them into a form acceptable by the *Solving Component*. The Solving Component solves a given planning problem by generating a day-ahead plan if such a plan exists.

Realising the planning system

We follow two more lifecycle phases to implement the planning system: Planning Tools Selection and Implementation (Georgievski 2023). First, we search existing planning technology for suitable tools. To the best of our knowledge, no existing tools implement the Day-Ahead Pricing and Problem Generation Components, considering our requirements. We can look at existing temporal planners for potential candidates for the Parsing and Solving Components, as most planners offer parsing PDDL planning problems *coupled* with solving the problems. Second, we implement the rest of the components in Python and perform domain modifications if needed.

Planning tools selection

One can look for planners in the International Planning Competition (IPC), which the AI planning research community organises to showcase the most advanced planners in various benchmark planning domains. The planners in the 2004 edition of IPC seem to be considered the most powerful temporal planners available (Hoffmann and Edelkamp 2005). Unfortunately, none of those planners can handle TIFs. Two other temporal planners, POPF2 (Coles et al. 2010) and UPMurphi (Penna et al. 2009), can deal with TIFs by treating them as an extension of TILs, however, imposing some limitations (Piacentini et al. 2015). Finally, POPF-TIF is a temporal planner that can handle TIFs and overcome such limitations (Piacentini et al. 2015). So, our choice of the temporal planner is substantially limited to a single planner. In a nutshell, POPF-TIF is a proof-of-concept extension of POPF2 that can maintain a partial order over actions in plans and manage deadlines. POPF-TIF also has some limitations. Due to the large search space, it faces difficulties when solving problems with complex temporal constraints in terms of longer planning times or suboptimal plans. Performance issues can also arise when the planning domain contains many actions, making it hard for the planner to explore all possibilities. Also, POPF-TIF cannot deal with negative preconditions, which require additional predicates, thus increasing the number of predicates and potentially affecting the plan accuracy (Benton et al. 2012).

Implementation

As an external energy market, we use the European Network of Transmission System Operators for Electricity (ENTSO-E), representing the transmission system operators of 35 European countries and providing day-ahead energy prices (ENTSO-E Assembly 2023). To obtain day-ahead prices from the ENTSO-E, we sent a GET request to its Application Programming Interface (API) endpoint, including an API key, start and end dates, and market area. The API responds with a JavaScript Object Notation (JSON) object upon which we do the filtering and binning.

The Problem Generation component takes the day-ahead prices, retrieves environmental data from local storage, and generates a problem instance file in PDDL. Specific to this component is that it divides a problem instance into two sub-instances based on time periods (one for 00:00-5:00 and another for 15:00-24:00). This division is necessary because of the performance issues of POPF-TIF. The component then passes the two sub-instances and the planning domain file to POPF-TIF.

Recall that POPF-TIF does not support negative preconditions. Therefore, we need to reconfigure the planning domain by replacing negated predicates (e.g., `(not (turned-on cihl-dishwasher))`) in preconditions with positive predicates that indicate the negation (e.g., `(turned-off cihl-dishwasher)`). Also, since the planner must ensure a time horizon of 24 h and that all constraints remain within the bound of that time horizon, it needs an *envelope action*. This domain-exogenous action is the first one in the plan and guarantees all other plan actions operate within the bounds specified by the envelope. It asserts a condition specified by every other plan action, ensuring other actions do not start before the envelope opens and helps prevent actions from exceeding the constraints defined by the bounds (Piacentini et al. 2015). This action

```
0.00000: (isend) [14.09500]
0.00100: (battery_charge_low_price) [2.00000]
0.00100: (out_of_operating_hours_all_off_lightsreduced) [4.00000]
2.00200: (battery_charge_low_price) [2.00000]
4.00200: (out_of_operating_hours_all_off_lightsreduced) [4.00000]
4.00300: (battery_charge_low_price) [2.00000]
6.00400: (battery_charge_low_price) [2.00000]
8.09100: (hvac_nominalprice_occupied_intemprange) [2.00000]
8.09200: (battery_discharge_nominalprice) [2.00000]
8.09200: (nominalprice_uncontrollable_loads_controllable_independent_loads) [2.00000]
8.09200: (ordinary_lights_nominalprice_occupied_highnaturallight) [2.00000]
10.09200: (hvac_nominalprice_occupied_intemprange) [2.00000]
10.09300: (ordinary_lights_nominalprice_occupied_highnaturallight) [2.00000]
10.09300: (battery_discharge_nominalprice) [2.00000]
10.09300: (nominalprice_uncontrollable_loads_controllable_independent_loads) [2.00000]
12.09300: (hvac_nominalprice_occupied_intemprange) [2.00000]
12.09400: (battery_discharge_nominalprice) [2.00000]
12.09400: (nominalprice_uncontrollable_loads_controllable_independent_loads) [2.00000]
12.09400: (ordinary_lights_nominalprice_occupied_highnaturallight) [2.00000]
13.90100: (isend) [10.10100]
14.00100: (hvac_nominalprice_occupied_intemprange) [2.00000]
14.00200: (battery_discharge_nominalprice) [2.00000]
14.00200: (nominalprice_uncontrollable_loads_controllable_independent_loads) [2.00000]
14.00200: (ordinary_lights_nominalprice_occupied_highnaturallight) [2.00000]
17.00100: (ordinary_lights_highprice_occupied_lownaturallight) [2.00000]
17.00100: (highprice_uncontrollable_loads_controllable_independent_loads) [2.00000]
17.00100: (battery_discharge_highprice) [2.00000]
17.00100: (hvac_highprice_occupied_intemprange) [2.00000]
19.00200: (battery_discharge_highprice) [2.00000]
20.00100: (out_of_operating_hours_all_off_lightsreduced) [4.00000]
```

Fig. 3 Example of a day-ahead plan

must not end until the problem goal is achieved, ensuring all plan actions have the time to execute and achieve their objectives. Our constraints for the envelope action include maintaining the current temperature within the optimum temperature range and ensuring the battery charging and discharging limits remain inside the envelope. With this, POPF-TIF computes the best possible day-ahead plan that minimises energy cost and consumption while not compromising environmental conditions. Figure 3 shows a plan example.

Evaluation

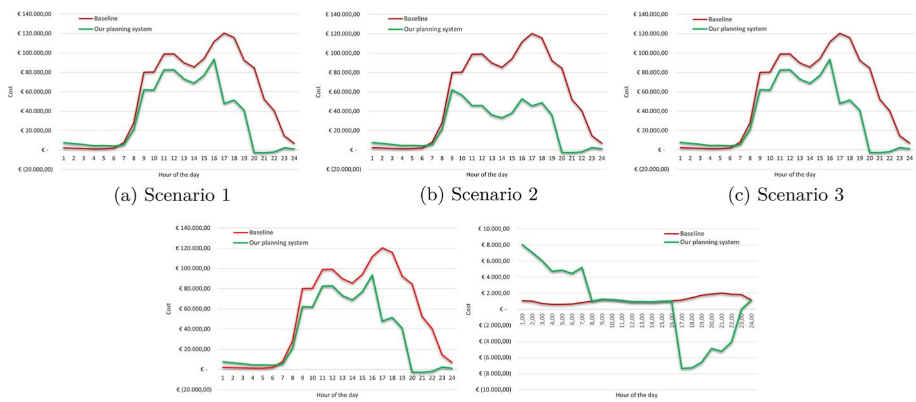
We want to evaluate our planning system in terms of the energy cost of the resulting day-ahead plans in several scenarios that can occur in a building and compare that cost with the energy cost of the building without using our system, that is, the baseline. We present the experimental design next, followed by the results.

Experiment design

The experiment design has two main components, an energy profile of a real building and a set of scenarios simulating building situations. A building energy profile is a detailed record of the energy consumed by each load within the building, typically over a day. Given our comparison objective, we need a building energy profile to determine each load's energy consumed and associated costs. We have obtained an energy profile of a medium-sized commercial building in the United States of America from the Office of Energy Efficiency and Renewable Energy (U.S. Depart. of

Table 1 Scenarios settings and their energy costs

Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Day	Weekday	Weekday	Weekday	Weekday	Weekend
Occupancy	7 hours (High)	4 hours (Low)	7 hours	7 hours	No
Natural light level	High	High	Low	High	Normal
Temperature in range	Yes	Yes	Yes	No	Off
Baseline cost (€)	1 313 108.74	1 313 108.74	1 313 108.74	1 313 108.74	27 869.02
Our cost (€)	798 755	558 162.54	807 773.26	888 250.61	15 432.86
Cost reduction	39%	57%	38%	32%	44.62%



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

(d) Scenario 4 (e) Scenario 5

Fig. 4 Scenario's energy cost with and without the use of our planning system

Energy 2023). We have selected two days from the profile (i.e., weekday and weekend) and calculated the energy cost based on day-ahead prices for the date of February 6th, 2023.

We also need data related to environmental conditions. We opt for simulating environmental conditions as such data is unavailable for the selected building. In particular, we design five scenarios by varying the duration of overall building occupancy, outside natural light, and current temperature to observe the impact of each condition on the energy cost. The upper part of Table 1 summarises the five scenarios. Scenario 1 is about a weekday and defines optimum conditions: the occupancy is set to 7 h, which is the maximum occupancy, the outside natural light level is good, and the current temperature is within the optimum range. Scenario 2 is also about a weekday under reduced occupancy: the occupancy is set to 4 h, which is the minimum occupancy. Scenario 3 is about a weekday under lousy light conditions: the outside natural light level is set to low, the occupancy is set to its maximum, and the current temperature is at its optimum. Scenario 4 is also about a weekday where the current temperature is out of its optimal range—the outside temperature is low as in a winter season, causing the building temperature to decrease. All other parameters are the same as in Scenario 1. Finally, Scenario 5 is about a weekend day, where all considered loads except for the ordinary lights are off. The occupancy is set to zero, and the outside light level is set to good. The current temperature is unnecessary as all loads that depend on it are switched off.

Results

Table 1 also shows the energy-cost reduction per scenario. On average, our planning system can reduce the energy cost by 43% compared to the baseline energy cost. In Scenario 1, the energy cost for the day before using our planning system is way above one million euros, while with our system, it decreases to a little less than eight hundred thousand euros—the system achieved a cost reduction of 39%.

Figure 4a shows the hourly cost savings the planning system achieved compared to the baseline in Scenario 1. In the first seven hours, the cost of the day-ahead plan is higher than the baseline, as the battery is charging at a low energy price, consuming more energy and resulting in higher costs. However, this is compensated between 18:00 and 23:00 when there is a high energy price, significantly reducing energy costs. Scenario's day-ahead plan reduces cost by 57%, from over a million euros to about half a million euros, compared to the baseline's energy cost. This is 18% more reduction than the one achieved under Scenario 1's optimum conditions. Figure 4b shows the hourly cost savings achieved in Scenario 2. The substantial energy cost decrease results from reduced energy consumption due to lower occupancy. This can be attributed to the change in two loads: ordinary lights and HVAC. The day-ahead plan for Scenario 3 produced a significant cost reduction of 38%. It shows the system can achieve energy cost savings even under sub-optimal environmental conditions. The cost decrease can be attributed to the increased usage of ordinary lights, which increased energy consumption during hours of low natural light levels. Figure 4c depicts the hourly energy cost savings achieved in Scenario 3. In Scenario 4, the energy cost with our planning system is almost nine hundred thousand euros, a reduction of 32% under sub-optimal conditions. Figure 4d shows the hourly energy cost for Scenario 4. The energy cost is significantly higher during the building's operating hours due to the low outside temperature, which requires heating. However, the planning system can compensate for the cost increase later in the day and greatly reduce costs despite the sub-optimal conditions. Figure 4e shows the hourly energy cost for Scenario 5. There is no occupancy, and all loads are turned off during weekends, so the energy cost is significantly lower than on weekdays. Our planning system utilised this opportunity to charge the battery during the low-price period (00:00–07:00) and discharge it during the high-price period (17:00–23:00), resulting in a remarkable cost reduction of 44.62%.

Discussion and conclusion

We engineer a planning system for operating commercial buildings based on AI temporal planning that uses environmental conditions and day-ahead energy prices to compute plans that can positively impact buildings' energy costs. We base the system design on the software development life cycle for engineering AI planning systems to ensure having a well-defined and systematically designed system. It is also advantageous to support the system's extendability and move to deployment.

A crucial system element is the planning domain, which we model based on domain-oriented requirements. While the discussed requirements are relatively broad and general, the planning domain incorporates knowledge about *selected* elements (e.g., ordinary lights). Moreover, our primary modelling assumption is that each element has a single instance, which may not coincide with reality. This can limit the system's applicability, although no obstacles prevent one from extending the planning problem with more elements and instances. We made other domain design decisions to keep the modelling process uncomplicated (e.g., the action number per load) and pragmatic (e.g., a fixed duration of actions). These decisions may affect the quality of the planning domain model and the performance of the planning system in terms of the ability to find plans and plan optimality. It is necessary to perform additional experiments to assess the domain's quality and impact.

When considering building environment data, we assume that our planning system can get the data all at once. While this demonstrates the planning system's effectiveness, the Problem Generation Component needs to be further enhanced and refined to integrate with a gateway of sensors and dynamically accept sensor data whenever a relevant change in the building environment occurs.

We assumed the initial state of the temporal planning problem is fully observable and actions are deterministic. While the Problem Generation Component takes in the most recent data about a building's environment, the Solving Component generates plans in an offline mode—it does not consider environmental changes that may happen during plan computation. Further extensions and measures are needed to account for uncertainty and unexpected events in dynamic building environments (see, for example, our previous work on handling action contingencies (Kaldeli et al. 2016)).

Recall that we divided our problem instance into two parts. The division decision was enforced because the selected planner could not complete the computation on a full problem instance (we ran the planner for several hours). These performance issues may prevent us from day-ahead planning for an entire day. On the other hand, the selected planner could compute plans for the divisions in about 7 s. Plan generation is theoretically proven to be expensive in terms of time and space, and temporal planning problems are known to be EXPSPACE-complete (see Rintanen 2007).

We evaluated the planning system using scenario analysis involving simulated data and compared it with a baseline. We demonstrated that the planning system could reduce the energy costs by 43% on average and 56% at maximum when compared to the baseline cost. These results showcase the potential of AI planning, particularly temporal planning, to significantly and positively impact the operational costs of buildings. Evaluating the system using realistic data should not drastically affect the results showing our approach's effectiveness. One could argue our planning system is expectedly more advantageous than the system that produced the baseline of the commercial building. While this is not a threat to the applicability of our system, it can be a potential warning about the system's benefits compared to other advanced solutions. Further research is needed to understand how our approach based on temporal planning compares to other advanced approaches. However, we are optimistic given the encouraging evidence of using AI planning in actual buildings (Georgievski et al. 2017).

Author contributions

IG conceived the presented idea, contributed to the system design, verified the results, and wrote the manuscript. MZK designed, developed, and evaluated the system under the supervision of IG. MA discussed the results, reviewed, and finalised the manuscript. All authors read and approved the final manuscript.

About this supplement
This article has been published as part of Energy Informatics Volume 6 Supplement 1, 2023: Proceedings of the 12th DACH+ Conference on Energy Informatics 2023. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-6-supplement-1.

Availability of data and materials
Software and data developed in this study are available from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Published: 19 October 2023

References
Aiello M, Fiorini L, Georgievski I (2021) Software engineering smart energy systems. In: Fathi M, Zio E, Pardalos PM (eds.) Handbook of Smart Energy Systems, Springer, Cham, pp 1–21.
Bajada J (2016) Temporal planning for rich numeric contexts. PhD thesis, Dept. of Informatics School of Natural and Mathematical Sciences, King's College
Benton J, Coles A, Coles A (2012) Temporal planning with preferences and time-dependent continuous costs. In: International Conference on Automated Planning and Scheduling, pp 2–10
BMWi: energy efficiency strategy for buildings (2015) https://bit.ly/3hJy9Ch. Accessed 23 Jun 2023
BMWk: ordinances on saving energy (2022) http://bit.ly/44fwZVD. Accessed 23 Jun 2023
Coles A, Coles A, Fox M, Long D (2010) Forward-chaining partial-order planning. In: International Conference on International Conference on Automated Planning and Scheduling, pp 42–49
Edelkamp S, Hoffmann J (2004) PDDL2. 2: the language for the classical part of the 4th international planning competition. Tech. Rep. 195, University of Freiburg
ENTSO-E assembly: European Network of TSO. https://www.entsoe.eu/. Accessed 23 Jun 2023
Fox M, Long D (2003) Pddl2.1: an extension to PDDL for expressing temporal planning domains. J Artif Intell Res 20(1):61–124
Georgievski I (2023) Conceptualising software development lifecycle for engineering AI planning systems. In press
Georgievski I (2023) Software development lifecycle for engineering AI planning systems. In: International Conference on Software Technologies, pp 751–760
Georgievski I, Aiello M (2022) Temporal planning for building automation. In: International Conference on Applications of Intelligent Systems. Extended abstract
Georgievski I, Nguyen TA, Nizamic F, Setz B, Lazovik A, Aiello M (2017) Planning meets activity recognition: service coordination for intelligent buildings. Pervasive Mob Comput 38(1):110–139
Ghallab M, Nau DS, Traverso P (2004) Automated planning: theory & practice. Morgan Kaufmann Publishers Inc., San Francisco
Hoffmann J, Edelkamp S (2005) The deterministic part of IPC-4: an overview. J Artif Int Res 24(1):519–579
IEA: evolution of energy prices, 2020-2021 (2021) https://bit.ly/43U5uBi. Accessed 23 Jun 2023
IEA: buildings (2022) https://www.iea.org/reports/buildings. Accessed 23 Jun 2023
Kaldeli E, Warriach EU, Lazovik A, Aiello M (2013) Coordinating the Web of services for a smart home. ACM Tran. on the Web 7(2)
Kaldeli E, Lazovik A, Aiello M (2016) Domain-independent planning for services in uncertain and dynamic environments. Artif Intell 236(C):30–64
McDermott D, Ghallab M, Howe A, Knoblock C, Ram A, Veloso M, Weld DS, Wilkins D (1998) PDDL –the planning domain definition language. Tech. Rep. CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control
Penna GD, Magazzeni D, Mercorio F, Intrigila B (2009) UPMurphi: a tool for universal planning on PDDL+ problems. In: International Conference on Automated Planning and Scheduling, pp 106–113
Piacentini C, Alimisis V, Fox M, Long D (2015) An extension of metric temporal planning with application to AC voltage control. Artif Intell 229:210–245
Piacentini C, Fox M, Long D (2015) Planning with numeric timed initial fluents. In: AAAI Conference on Artificial Intelligence, AAAI Press, Austin, Texas, pp 4196–4197
Rintanen J (2007) Complexity of concurrent temporal planning. In: International Conference on International Conference on Automated Planning and Scheduling, pp 280–287
Shaikh A (2021) Energy Optimization in HVAC Using Automated Temporal Planning. Master's thesis, Dhaka University Of Engineering & Technology
U.S. Department of Energy: commercial reference buildings. https://bit.ly/3PuAyDk. Accessed 23 Jun 2023
University of Stuttgart: save energy. Together! https://bit.ly/3NRLvNZ. Accessed 23 Jun 2023

Publisher's Note