

RESEARCH

Open Access



Optimization of district heating production with thermal storage using mixed-integer nonlinear programming with a new initialization approach

Jakob Bjørnskov*, Lasse Kappel Mortensen, Konstantin Filonenko, Hamid Reza Shaker, Muhyiddine Jradi and Christian Veje

From 1st Energy Informatics.Academy Conference Asia
Beijing, China . 29-30 May 2021

* Correspondence: jabj@mmmi.sdu.dk

Center for Energy Informatics,
Maersk Mc-Kinney Moller Institute,
University of Southern Denmark,
Campusvej 55, DK-5230 Odense M,
Denmark

Abstract

Non-convex scheduling of energy production allows for more complex models that better describe the physical nature of the energy production system. Solutions to non-convex optimization problems can only be guaranteed to be local optima. For this reason, there is a need for methodologies that consistently provide low-cost solutions to the non-convex optimal scheduling problem. In this study, a novel Monte Carlo Tree Search initialization method for branch and bound solvers is proposed for the production planning of a combined heat and power unit with thermal heat storage in a district heating system. The optimization problem is formulated as a non-convex mixed-integer program, which is incorporated in a sliding time window framework. Here, the proposed initialization method offers lower-cost production planning compared to random initialization for larger time windows. For the test case, the proposed method lowers the yearly operational cost by more than 2,000,000 DKK per year. The method is one step in the direction of more reliable non-convex optimization that allows for more complex models of energy systems.

Keywords: Mixed integer non-linear optimization, MINLP, District heating scheduling, Modeling, Tree search, Initialization, Non-convex optimization

Introduction

Scheduling of district heating production is a well-studied problem in literature (Deng et al. 2017; Lésko et al. 2018; Gopalakrishnan and Kosanovic 2015; Rong and Lahdelma 2007). It is vital to the daily operation of district heating systems in order to keep operational costs low and ensure proper functioning to meet demand at all times. Additionally, the optimal scheduling problem is used for techno-economic assessment of new potential investments and when building new District Heating (DH) systems



© The Author(s). 2021 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

(Elsido et al. 2017a). Generally, there are four formulations of the optimal scheduling problem, linear programming (LP) (Lozano et al. 2009; Rong and Lahdelma 2005), mixed-integer linear programming (MILP) (Söderman and Pettersson 2006; Arcuri et al. 2007), non-linear programming (NLP) (Bindlish 2016) and mixed-integer non-linear programming (MINLP) (Deng et al. 2017; Lésko et al. 2018). The advantage of MILPs and LPs is that they can be solved with commercially available solvers for a global optimum. Despite this feature, using linear models to describe physical systems that typically are highly non-linear introduces error into the model, as a linear model of a non-linear system can at best only be a good approximation. Non-linear models on the other hand, while allowing better systems descriptions are inherently more difficult to solve. For non-linear models that are also non-convex, the problem is even worse, because a solution to a non-convex problem can only be guaranteed to be locally optimal. A common technique for dealing with mixed-integer non-linear models is to approximate the models as mixed-integer linear programs via linearization in order to guarantee that the solution is globally optimal (Lésko et al. 2018; Elsidio et al. 2017b). The accuracy of a linear approximation is dependent on how many linear segments a non-linear function is divided into. This introduces a trade-off between the complexity and the accuracy of the approximation. The more segmented the piece-wise linearization is the better the accuracy. However, each segment introduces additional variables to the model which makes the model bigger and more complex.

As an alternative to linear approximation, several methods exist for solving mixed-integer non-linear programs which do not rely on linearizing the problem. The methods can generally be divided into two groups, derivative-based and derivative-free. Derivative-free algorithms for mixed-integer non-linear programs include evolutionary, genetic, swarm intelligence algorithms (Elsido et al. 2017b; Boukouvala et al. 2016; Luo et al. 2007). Derivative-based methods employed in commercially available software for MINLP include cutting planes, branching and bounding (Boukouvala et al. 2016). As the solution of non-convex optimization problems is not guaranteed to be globally optimal, a common technique to find a good local solution is to repeatedly solve the optimization problem (Tveit et al. 2009; Savola et al. 2007). The employment of this method benefits from computationally efficient solvers and methods that consistently yield solutions with a low optimality gap.

In a recent study (Makkonen and Lahdelma 2006), the authors propose the use of the Power Simplex branch-and-bound algorithm to solve the non-convex scheduling problem of a combined heat and power unit operation. The problem is divided into hourly subproblems, which can be solved sequentially. This is a feasible choice in part because the systems do not have a thermal energy storage unit. The authors emphasize that the Power Simplex branch-and-bound algorithm is efficient because of its capability to reuse parent nodes to calculate child nodes. Rong and Risto (Rong and Lahdelma 2007) proposes an envelope-based branch-and-bound algorithm that employs a pruning technique to improve the computation speed of solving non-convex non-linear mixed-integer program for production planning of a combined heat and power plant. Similar to Simo and Risto (Makkonen and Lahdelma 2006) the problem is formulated as on hourly optimization. The authors succeed in decreasing the computation time significantly compared to ILOG CPLEX 9.0 MIP solver and the Power Simplex branch-and-bound solver. Gopalakrishnan and Kosanovic (Gopalakrishnan and Kosanovic 2015)

proposes a hybrid genetic algorithm for the solution of the non-convex optimal scheduling problem of a combined heat and power plant. It combines genetic algorithms for exploring the integer solutions space and employs a gradient-search for exploiting isolated regions of the solution space. The authors highlight that the proposed algorithm outperforms classical branch-and-bound algorithms in terms of capability to find integer feasible solutions and moreover that the optimality gap of solutions with the proposed method is lower than with branch-and-bound. Lastly, the authors emphasize that the proposed methods are computationally more efficient than classical branch-and-bound algorithms.

In two studies (Tveit et al. 2009; Savola et al. 2007), the repeated initialization of the non-convex solver is based on random initial solutions. This method is intuitively good at probing the solution space, but its speed can be questioned. To improve the computational efficiency of solving non-convex mixed-integer non-linear problems, (Soares et al. 2015) proposes a warm start method in combination with the Differential Search and the Quantum Particle Swarm Algorithms. The warm start methods are based on the solution of the convex relaxation of the non-convex problem. The authors found that the warm start method combined with evolutionary and swarm intelligence algorithms is capable of drastically reducing computation time.

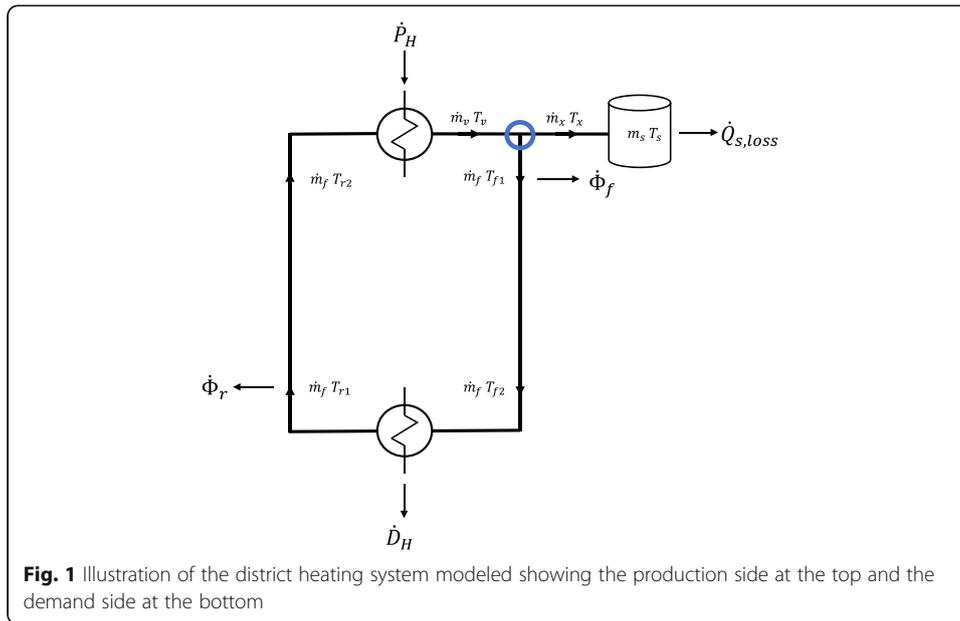
This paper proposes a new warm start initialization procedure based on a stochastic discrete tree search, called Monte Carlo Tree Search (MCTS), which constructs initial feasible solutions for a multi-period steady-state scheduling problem in a DH system with a combined heat and power unit and thermal storage. The authors are not aware of any other studies where tree search has been used as an initialization method. The proposed method is more consistent in finding good solutions than random initialization when the problem size grows. Therefore, the method constitutes an improvement over random initialization when planning for longer periods (12–14 h), which enables smarter storage use and lower operational cost.

The paper is structure as follows; 1) a mathematical model is first presented which describes the district heating system for which the scheduling problem is solved, 2) an optimization problem is defined based on the developed model, 3) the new Monte Carlo Tree Search initialization methods is introduced, 4) the experimental setup is defined, 5) the results are presented and discussed and 6) a conclusion is made.

District heating system model development

In this section, a mathematical model of a DH system consisting of a combined heat and power unit (CHP) with Thermal Energy Storage (TES) is developed. The model will briefly be presented, but not explained in detail as the model is simply regarded as a test suite for the optimization methods tested. The system is modeled as a quasi-dynamic system with a time step of 1 h and it is illustrated in Fig. 1.

The production-side is located on the upper half of the figure where the plant supplies thermal energy \dot{P}_H to the system given by 1). The plant operator then has the option to either charge or discharge the TES resulting in the thermal energy flow \dot{Q}_{flow1} to or from the pipe node given in (2). The node has been marked with a blue ring on the figure. The direction of the flow depends on the binary variable x_i . (3) gives the thermal energy supplied to the transmission line while (4) and (5) are the energy and mass



balance equations for the node respectively. Lastly, (6) describes the temperature of the water flowing to or from the storage, which also depends on the flow direction.

$$\dot{P}_{H,i} = \dot{m}_{v,i} C_{p1} (T_{v,i} - T_{r2,i}) \tag{1}$$

$$P_{H,i} - Q_{flow1,i} - Q_{flow2,i} = 0 \tag{2}$$

$$\dot{Q}_{flow2,i} = \dot{m}_{f,i} c_{p,1} (T_{f1,i} - T_{r2}) \tag{3}$$

$$\dot{Q}_{flow1,i} = \dot{m}_{x,i} c_p (2x_i - 1) (T_{x,i} - T_{r2,i}) \tag{4}$$

$$\dot{m}_{v,i} - \dot{m}_{x,i} (2x_i - 1) - \dot{m}_{f,i} = 0 \tag{5}$$

$$T_{x,i} = T_{v,i} x_i + T_{s,i} (1 - x_i) \tag{6}$$

The heat loss per increment pipe is proportional to the temperature difference between fluid temperature and ground temperature. For simplicity, it is assumed that the fluid temperature is constant during the whole pipe segment and only drops at the outlet. This gives the forward and return heat loss expressed in (7) and (8) respectively. Eqs. (9), (10) and (11) describe the temperature of the water flow in terms of the forward heat loss $\Phi_{f,i}$ the heat demand \dot{D}_H and the return heat loss $\Phi_{r,i}$ respectively. The $U_{par}L$ factor has been derived empirically to match data for one of the transmission lines of an actual CHP plant. Lastly, the pump work of the system is given in (12) as a third-order polynomial, which also has been derived empirically from data for a transmission line at a district heating company.

$$\Phi_{f,i} = (T_{f1,i} - T_{soil}) U_{par} L \tag{7}$$

$$\Phi_{r,i} = (T_{r1,i} - T_{soil}) U_{par} L \tag{8}$$

$$T_{f2,i} = T_{f1,i} - \frac{\Phi_{f,i}}{\dot{m}_{f,i}c_{p1}} \quad (9)$$

$$T_{r,i} = T_{r2,1} - \frac{\dot{D}_{H,i}}{\dot{m}c_p} \quad (10)$$

$$T_{r2,1} = T_{r1,i} - \frac{\Phi_{r,i}}{\dot{m}_{f,i}c_p} \quad (11)$$

$$\dot{W}_p = \alpha_3 \dot{m}_{f,i}^3 + \alpha_2 \dot{m}_{f,i}^2 + \alpha_3 \dot{m}_{f,i} \quad (12)$$

The TES is modeled as a lumped body meaning that the temperature is assumed uniform across the entire volume. The energy balance of the storage $Q_{s,i}$ is expressed in (13). The thermal energy added or removed from the storage $\dot{Q}_{s,pipe,i}$ is given by (14) and the heat loss is given by Newtons Law of Cooling in (15) where the heat-conducting surface area $A_{s,i}$ is varying with the mass as given in (16). The mass balance of the storage is given in eq. (17) and the temperature of the storage is expressed in (18).

$$Q_{s,i} = Q_{s,i-1} + (\dot{Q}_{s,pipe,i} - \dot{Q}_{s,loss,i})\Delta t \quad (13)$$

$$\dot{Q}_{s,pipe,i} = \dot{m}_{x,i}c_p(2x_i-1)(T_{x,i}-T_R) \quad (14)$$

$$\dot{Q}_{s,loss,i} = hA_{s,i}(T_{s,i}-T_{a,i}) \quad (15)$$

$$A_{s,i} = A_{s,max} \frac{m_{s,i}}{m_{s,max}} \quad (16)$$

$$m_{s,i} = m_{s,0} + \sum_{j=1}^i \dot{m}_{x,j}(2x_j-1)\Delta t \quad (17)$$

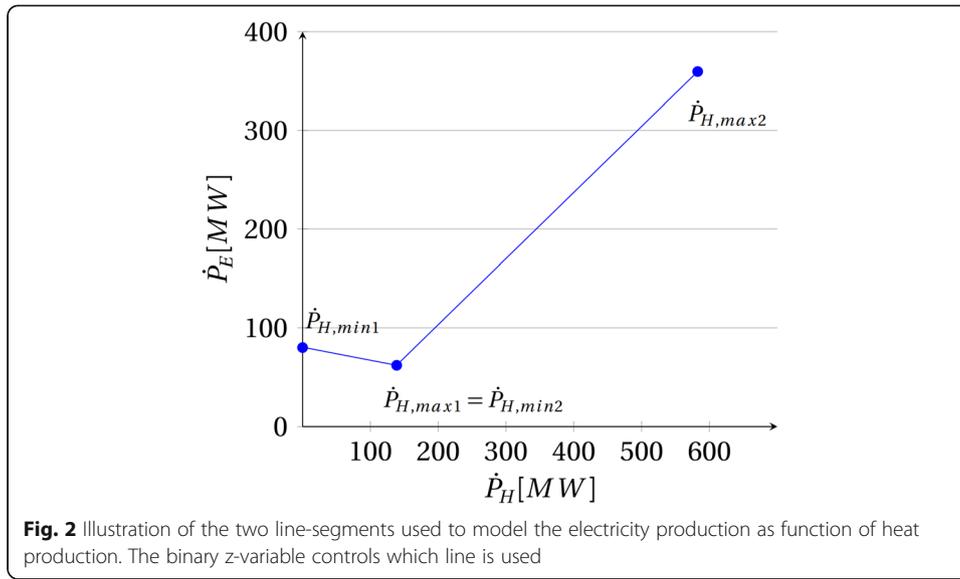
$$T_{s,i} = \frac{Q_{s,i}}{m_{s,i}c_{p2}} + T_R \quad (18)$$

The electricity produced by the CHP unit is modeled as shown in Fig. 2. The model is a simplified version of an actual CHP unit at a Danish district heating company. The figure shows that the plant has two modes of operation; $\dot{P}_{H,min1} \leq \dot{P}_H \leq \dot{P}_{H,max1}$ and $\dot{P}_{H,min2} \leq \dot{P}_H \leq \dot{P}_{H,max2}$. This is expressed mathematically in (19) by introducing the binary variable z .

$$\dot{P}_{E,i} = z_i(a_1\dot{P}_{H,i} + b_1) + (1-z_i)(a_2\dot{P}_{H,i} + b_2) \quad (19)$$

Optimization problem

The hourly cost of production is defined in (20), where the pump work, $\dot{W}_{p,i}$, and the electricity production, $\dot{P}_{E,i}$, is given in (12) and (19) respectively. EP_1 is the spot market electricity price and the second term, therefore, accounts for the revenue from the sale of electricity while the first term, accounts, for the fuel cost, where FP_i is the fuel price and \dot{F}_i is the fuel consumption given by (21). The objective function of the MINLP is thus defined by summing the costs for each time step in (22).



$$f_{c,i} = \dot{F}_i FP_i + (W_{P,i} - \dot{P}_{E,i}) EP_i \tag{20}$$

$$\dot{F}_i = \frac{\dot{P}_{E,i} + C_v \dot{P}_{H,i}}{\eta_E} \tag{21}$$

$$\text{Minimize } \sum_{i=1}^N f_{c,i} \tag{22}$$

In (21) an equivalent electricity production is calculated from the heat production by multiplying the heat production with the ratio between electricity and heat production at constant fuel consumption, C_v (The Danish Energy Agency and Energinet 2019). The production is then divided by the electrical efficiency η_E . The constraints for the optimization problem are:

$$m_{s, \min} \leq m_{s,i} \leq m_{s, \max}, m_{s,i} \in \mathbb{R}$$

$$T_{f2, \min} \leq T_{f2,i} \leq T_{f2, \max}, T_{f2,i} \in \mathbb{R}$$

$$T_{r1, \min} \leq T_{r1,i} \leq T_{r1, \max}, T_{r1,i} \in \mathbb{R}$$

$$T_{v, \min} \leq T_{v,i} \leq T_{v, \max}, T_{v,i} \in \mathbb{R}$$

$$\dot{m}_{v, \min} \leq \dot{m}_{v,i} \leq \dot{m}_{v, \max}, \dot{m}_{v,i} \in \mathbb{R}$$

$$\dot{m}_{x, \min} \leq \dot{m}_{x,i} \leq \dot{m}_{x, \max}, \dot{m}_{x,i} \in \mathbb{R}$$

$$0 \leq x \leq 1, x \in \mathbb{Z}$$

$$0 \leq z \leq 1, z \in \mathbb{Z}$$

$$z \dot{P}_{H, \min1} + (1-z) \dot{P}_{H, \min2} \leq \dot{P}_{H,i} \leq z \dot{P}_{H, \max1} + (1-z) \dot{P}_{H, \max2}, \dot{P}_{H,i} \in \mathbb{R}$$

The authors have decided to use a Branch-and-Bound solver named *Apopt* which is implemented in the *APMonitor Optimization Suite*. The optimization suite is free to

use, offers free cloud computing services, and is compatible with Matlab and Python (APMonitor 2020a). The Apopt solver uses a combination of an active set method and Branch-and-Bound to manage the integer variables (John et al. 2014; APMonitor 2020b). As steady-state optimization of large problems can be time-consuming, the authors have chosen to adopt a sliding time window method for dividing the optimal scheduling problem into smaller more manageable problems which can be solved in sequence. The framework for the optimization can be seen in Fig. 3. In this framework, the optimization of each window is solved Try_{max} times and the best solution to each subproblem is kept for the construction of the final solution.

The sliding time window method introduces two new hyperparameters, namely the length of the sliding window, W_L , and the stride, W_S , where $W_S \leq W_L$. If the stride is less than the window length it means, there is an overlap between every pair of adjacent windows. If this is the case, the suboptimization of the latter window takes precedence over the region of overlap. Another important consideration is that the minimum look ahead at any discrete time step is given by $W_L - W_S$. As an example, consider a sliding time window optimization with $W_L = 5$ and $W_S = 5$, in this example the planning of the 5th hour will not have accounted for any subsequent hours. If $W_S = 1$ instead, the scheduling at each discrete time step will have considered the subsequent 4 h.

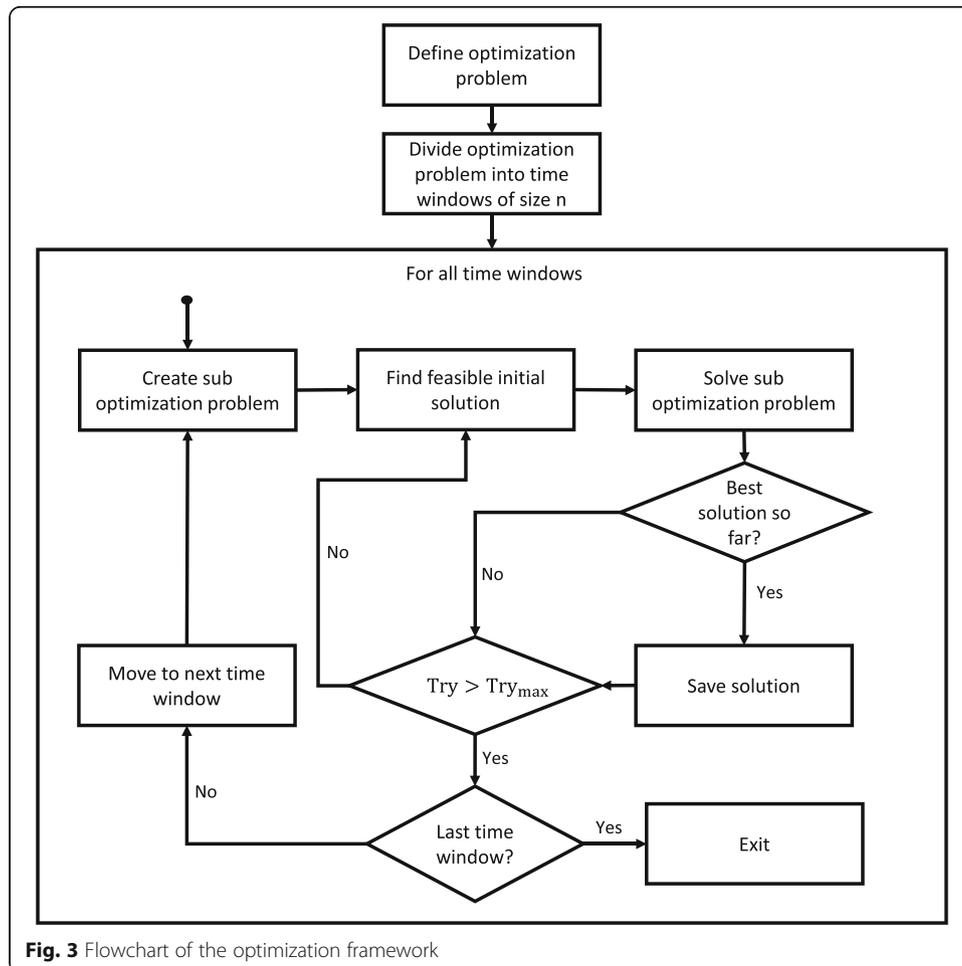


Fig. 3 Flowchart of the optimization framework

Monte Carlo tree search initialization method

The implementation of MCTS in this work requires a static set of actions. This set of actions is composed of several layers, one for each discrete time step in the optimization period. Each layer consists of a number of actions, which are sets of decision variables. E.g. for the optimization problem presented in this work, there are 5 decision variables chosen as $T_v, \dot{m}_v, \dot{m}_x, x, z$, which means an action could be defined as e.g. $a = \{60, 3, 1, 1, 0\}$. Given some state of the system $s_0 = \{m_s, 0, T_s, 0\}$, defined by the mass contained and the temperature of the storage, the action would then transition the system into a new state. Figure 4 visualizes an exemplified version of the actions available at each discrete time step. In the figure, each horizontal row is a layer consisting of several actions, shown as grey dots. The number of actions per time increment may be varied across time steps. Additionally, uniform noise is added to each action to ensure variance in the initial solution. It was found that increasing the number of discretizations for the variable \dot{m}_x improved the overall tree search for this problem. The static set of actions can be seen as a discrete mapping of the solution space. The task of the MCTS algorithm is therefore to search this set of actions to find good feasible candidate solutions in the discrete solution space.

When the set of actions for each hour has been created, it is deployed in an MCTS algorithm. The algorithm developed in this work is based on the Upper Confidence Bounds applied to Trees-algorithm (UCT) from (Kocsis and Szepesvári 2006) and the implementation in (Maddison et al. 2016). In this algorithm, each state-action pair (s, a) at simulation time t is associated with an expected reward $Q_t(s, a)$ and an exploration bias $u_t(s, a)$. Typically, the expected reward $Q_t(s, a)$ is calculated by averaging over all the backpropagated rewards that have passed through the node. Each reward is found in the selection phase when a leaf node is encountered. When this happens, random simulation is initiated from the selected leaf node until a terminal state is reached, determining the reward. The random simulation thus acts as a state evaluation function and is advantageous in situations where domain-specific evaluation functions are not available. This method was proved to guarantee an optimal policy when simulation time goes to infinity in (Kocsis and Szepesvári 2006). However, as argued in (Ramanujan and Selman 2011), in cases where a domain-specific evaluation function does exist, the search can be made more efficient and less time-consuming by replacing the random simulation with an evaluation function.

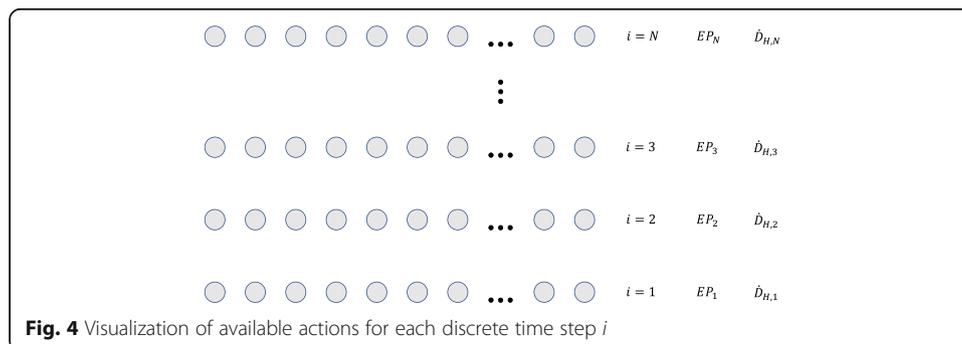


Fig. 4 Visualization of available actions for each discrete time step i

In the case of an optimization problem, an objective measure already exists in the form of an objective function, which justifies the before-mentioned replacement. For the implementation in this study, $Q_t(s, a)$ will also not represent a reward, but rather a cost as the problem solved is formulated as a minimization problem. Additionally, preliminary experiments in this work confirmed, that calculating $Q_t(s, a)$ based on the minimum value among the child nodes of state-action pair (s, a) , instead of averaging over backpropagated values, yielded better results as also discussed in (Ramanujan and Selman 2011). Therefore, $Q_t(s, a)$ is defined as shown in (23) while the exploration bias $u_t(s, a)$ is kept in the original form presented in (Kocsis and Szepesvári 2006), as given by (24).

$$Q_t(s, a) = \min_{b \in M(s_a)} [Q_t(s_a, b)] \tag{23}$$

$$u_t(s, a) = c \sqrt{\frac{\ln [\sum_{b \in M(s)} N_t(s, b)]}{N_t(s, a)}} \tag{24}$$

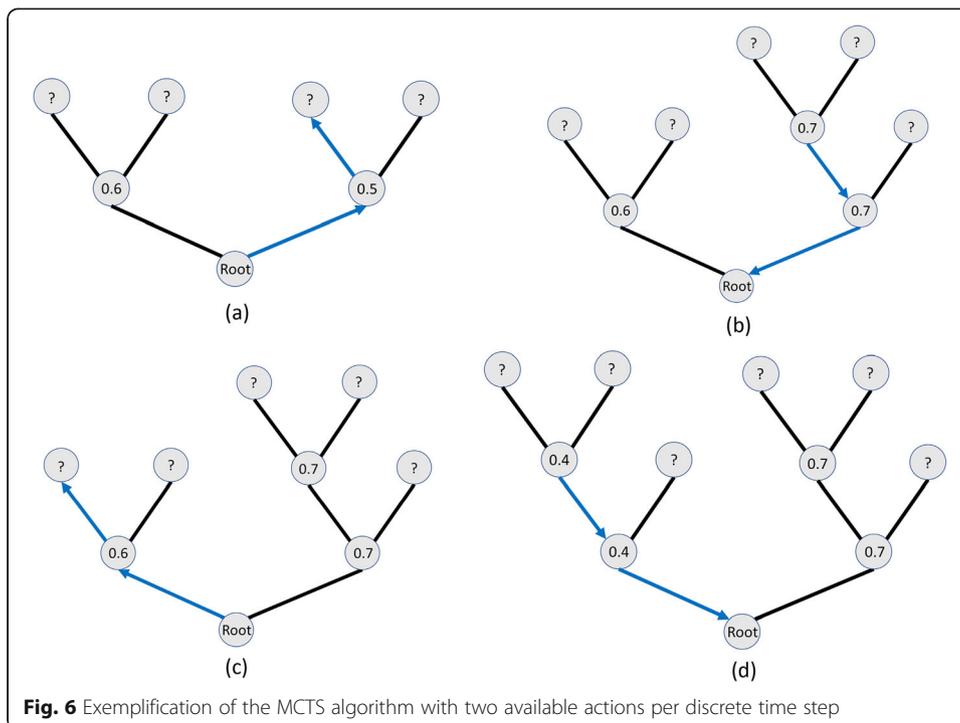
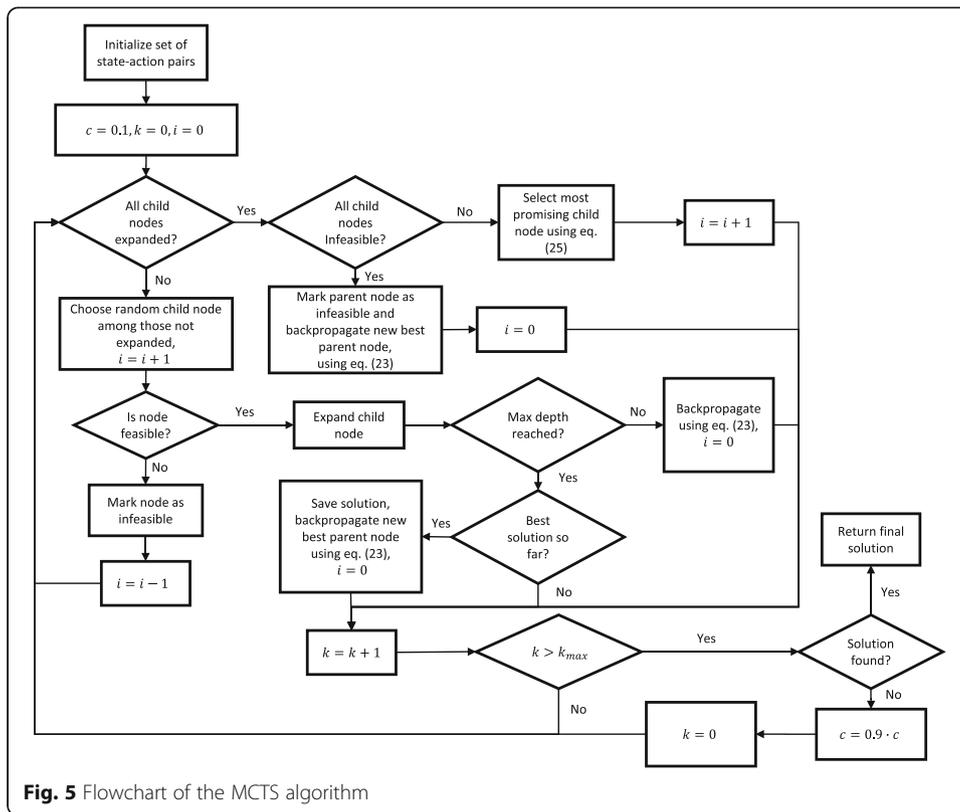
$$\pi(s) = \operatorname{argmin}_{b \in M(s_a)} [Q_t(s, b) - u_t(s, b)] \tag{25}$$

In (23), the expected cost $Q_t(s, a)$ of taking action a in state s is found by finding the minimum value among state-action pairs (s_a, b) for the resulting child state s_a . The set of actions available in state s_a is given by $M(s_a)$. In (24), $M(s)$ is the set of actions available in state s and $N_t(s, a)$ is the visit count of taking action a in state s . The equation, therefore, compares the total visit count of the parent state s to the child state resulting from action a . The function is designed such that it grows when an action is picked less than the alternatives and therefore encourages exploration. c is a hyperparameter controlling the trade-off between exploration and exploitation. The action selected in each state, called the *olicy* $\pi(s)$, is given by (25) based on both how promising the action looks; $Q_t(s, b)$, and the degree of exploration; $u_t(s, b)$. Figure 5 summarizes the algorithm in a flowchart.

To exemplify how the algorithm works, an illustration has been made in Fig. 6. The nodes drawn represent the system states while the edges represent actions transitioning the system from one state to another. As seen in Fig. 6a, where two nodes have already been expanded, the tree is traversed by iteratively selecting the best candidate among child nodes using (25). Each node stores two values in memory, $Q_t(s, a)$ and $N_t(s, b)$ which are updated each time backpropagation passes through the node. The selection is repeated until a child node is observed to also be a leaf node. In this case, a random child leaf node is then selected and expanded as shown in Fig. 6b. In this expansion, the feasibility of the chosen child node is first evaluated after which the scaled average objective value $Q_{t, leaf}$ is calculated and backpropagated through the parent nodes using (23). If the node is evaluated as infeasible, all child nodes can be pruned.

The average objective value is defined as the average objective of all the nodes traversed, including the frequently expanded node. This set of nodes is given as the set I in (26), where n is the number of nodes traversed. This is done to ensure comparability of the objective value when it is propagated back through the search tree.

$$\bar{f}_c = \frac{1}{n} \sum_{i \in I} f_{c,i} \tag{26}$$



$$Q_{t,leaf} = \frac{\bar{f}_c - f_{c, min}}{f_{c, max} - f_{c, min}} \tag{27}$$

When the average \bar{f}_c has been calculated, it is scaled to a value between 0 and 1 according to (27). Here, $f_{c, min}$ and $f_{c, max}$ are the pre-determined lower and upper bounds for the objective function. They are found by treating (20) as a linear program with the variables $W_{P, i}$, $P_{E, i}$ and $P_{H, i}$. The linear program is minimized and maximized separately for each of the two production modes in (19) for each hour $i \in \{1, \dots, N\}$. This results in $4N$ linear programming solutions among which the minimum and maximum values, $f_{c, min}$ and $f_{c, max}$, are extracted. This scaling makes it more convenient to tune the hyperparameter c in (24), as most use-cases of MCTS are within the domain of board-games where an outcome often is represented by 0 and 1 - loss and win. When the backpropagation reaches the root node, the selection starts over as shown in Fig. 6c, followed by expansion and backpropagation as shown in Fig. 6d.

Experimental setup

In order to test the effectiveness of the MCTS initialization method, it is incorporated in the optimization framework described in Fig. 3 as the “Find feasible initial solution”-step. It is bench-marked against random initialization using the Apopt Branch-and-Bound solver (BB). Additionally, the MCTS initial solution will be fed to the Apopt Non-Linear-Problem solver (NLP), which is the local solver used for each subproblem in the branch and bound algorithm. The three methods are tested on an optimization period of 48 h with a high variance in electricity price EP and a low variance in heat demand \dot{D}_H as shown in Fig. 7. As explained earlier in Fig. 3, this period is divided into several smaller optimization problems which are solved sequentially using the sliding time window approach. A stride of 5 is used for all simulations while the window

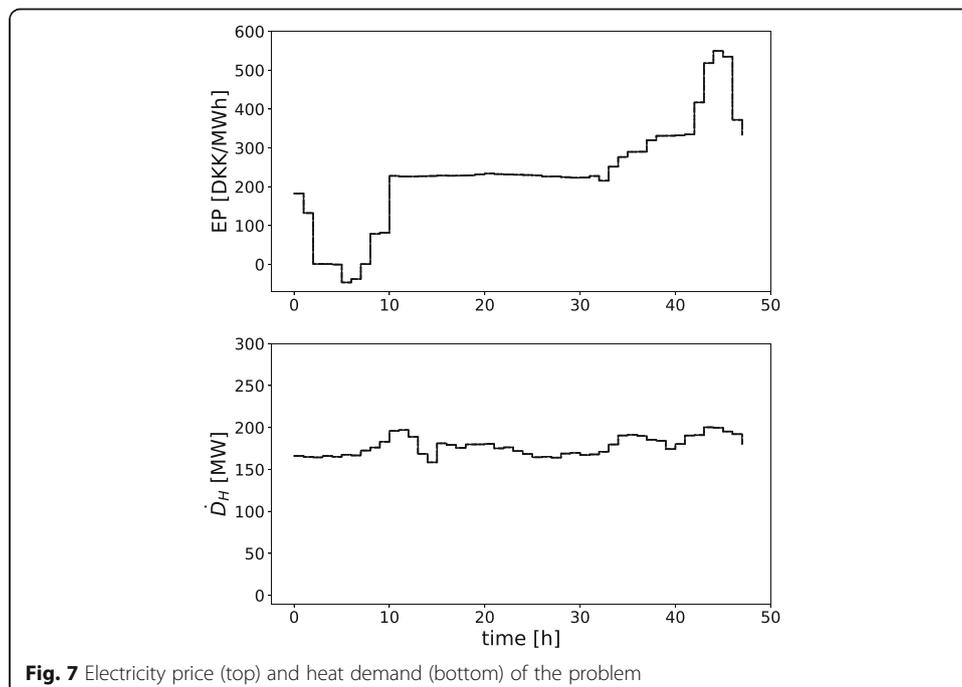


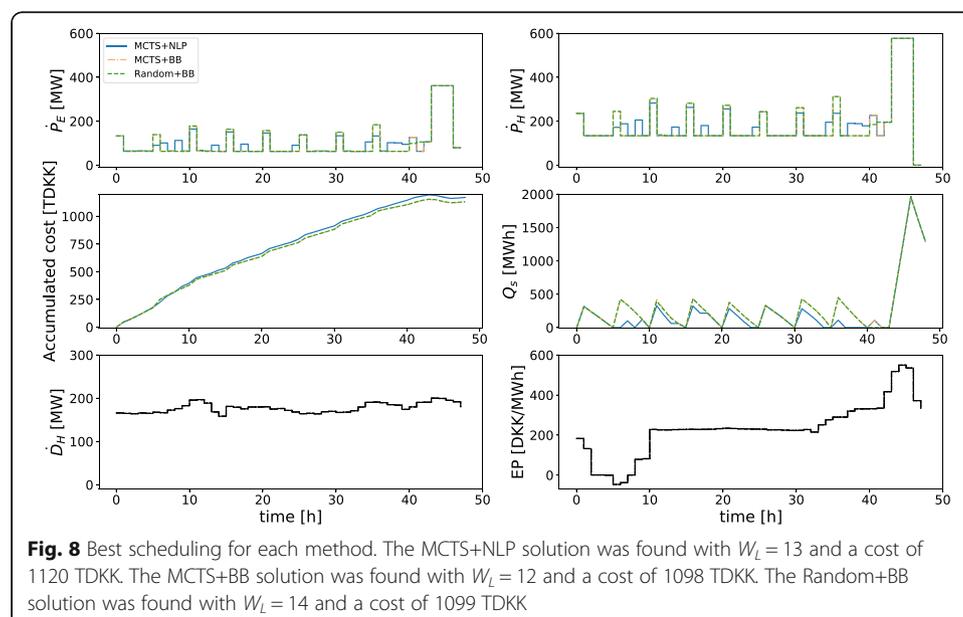
Fig. 7 Electricity price (top) and heat demand (bottom) of the problem

length W_L is varied from 5 to 14. Because of the stochastic nature of both MCTS initialization and random initialization, each sub-optimization is solved for 5 different initial solutions before moving the time window, i.e. $Try_{max} = 5$. The hyper-parameter c is initially set to 0.1 and is decreased by a factor of 0.9 for each $5e+4$ iteration until a solution is found after which c is held constant for additional $5e+5$ iterations. This is done to ensure that a solution is found within reasonable time and memory limits while still allowing for exploration. The whole framework illustrated in Fig. 3 is repeated 5 times for each time window length W_L , and for each of the 3 methods.

Results

Figure 8 shows the best operational schedule for each of the three methods. It is evident, that all methods yield operational plans that work to mitigate and or take advantage of fluctuations in the electricity spot price by using the TES to allow for overproduction of heat in hours with high electricity price and conversely to underproduce in hours with low electricity price, covering the heat deficit with stored thermal energy. Even though the best scheduling is markedly different for the three methods, the relative difference in the total cost is only approximately 2%. The best solution is found by the MCTS+BB method which gives a total cost of 1098 TDKK over the 48 h period. Comparing this solution with the worst obtained solution of 1199 TDKK, among all 45 simulations, instead gives a relative difference of approximately 9%. This gives a difference of 18.25 mDKK/year, which shows the importance of choosing a method that consistently produces low-cost operational plans.

To study the ability of each method to consistently produce low-cost operational plans, Fig. 9 is introduced, which shows boxplots of the cost and computation time of each method as a function of the size of the sliding time window. A key indicator of the consistency of a method in this presentation is the placement of the quartiles and the interquartile range. The interquartile range shows the spread in the operational cost of and time of the solutions of each method. The interquartile range of Random + BB



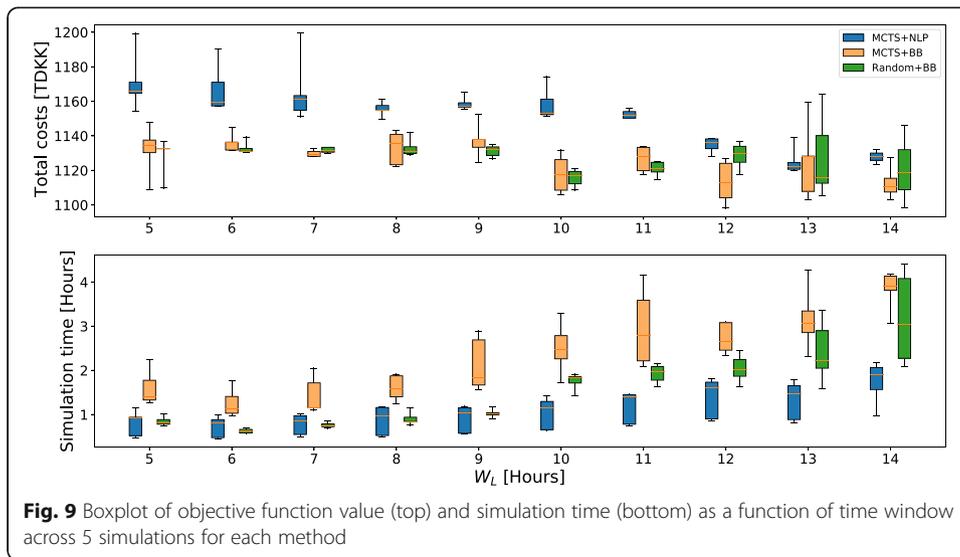


Fig. 9 Boxplot of objective function value (top) and simulation time (bottom) as a function of time window across 5 simulations for each method

increases as the time window increases, which is likely caused by the increase in the size of the solution space that makes it more difficult for the method to consistently yield low-cost solutions. Large time windows enable scheduling methods to make better use of the thermal storage to take advantage of fluctuations in the electricity price and heat demand, which is seen as the total cost decreases as the size of the sliding time window increases for all methods. A weakness of the Random+BB method is therefore that the consistency decreases with the time window size. MCTS + BB does not exhibit the same tendency, which indicates that it provides better consistency when the size of the solution space increases. This is more evident from Fig. 10, where the simulation results for windows in range 12 to 14 have been merged for the three methods. Here, the interquartile range is shifted down for MCTS + BB compared to Random + BB indicating that it is more feasible to use MCTS+BB for larger time windows. In fact, if the yearly cost is calculated assuming that the median cost of each method is applicable for the entire year, the cost of scheduling with MCTS+BB would be 2,182,700 DKK lower than Random+BB corresponding to 1.7 DKK/MWh.

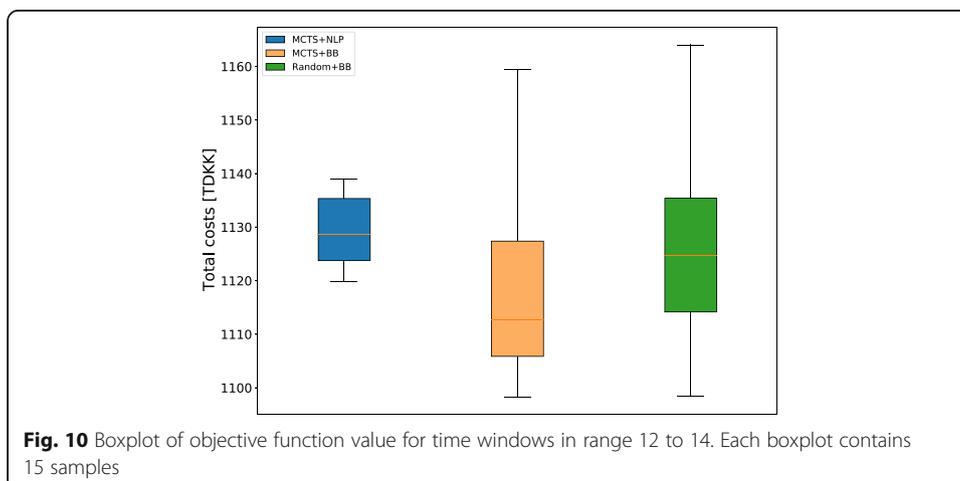


Fig. 10 Boxplot of objective function value for time windows in range 12 to 14. Each boxplot contains 15 samples

The improved consistency that MCTS + BB provides over Random + BB comes at the cost of computation time. The computation time is a vital factor in the optimization framework as the optimization of each time window is repeated to combat the non-convex nature of the MINLP. Figure 9 shows that the computation time of MCTS + BB is shifted upward by a somewhat constant amount compared to Random + BB. As the initialization is the only difference between the methods, the shift must be attributed to the MCTS algorithm. Despite of the longer computation time, MCTS + BB is a feasible method for scheduling district heating production. With the Nordic electricity market as a reference, the hourly spot price is only known 24 h in advance, which means that it is only feasible to plan for a total of 24 h ahead. This gives a computation time in the range of 1.5 to 2.0 h for the MCTS + BB method. This computation time can be lowered by parallelizing the optimizations of each time window. Also, the MCTS algorithm can be implemented more efficiently in a compiled language like C++ instead of Matlab to increase the speed of the algorithm. For improving the consistency of the method even further, the hyperparameter Try_{max} can be increased, so that the optimization of each time window is run more times. Implementing this change, it will be possible to get consistent low-cost production schedules while only running the overall framework of Fig. 3 once. The MCTS initialization in combination with a branch and bound solver thereby provides a concrete and effective tool for non-convex scheduling of district heating production.

Conclusion

This paper proposes an initialization method for branch and bound solvers to schedule district heating production with storage optimally. The method uses a tree search strategy to search a discretized solution space to find a low-cost region, used as an initial solution for the solver. The paper finds that the MCTS initialization method successfully improves the consistency of solutions compared with random initialization for larger window sizes under the application of the sliding time window approach. Making it a more scalable solution. MCTS lowered the per-unit cost of energy by 1.7 DKK/MWh compared to random initialization under the assumption that the difference between the median solutions of each would be constant throughout the year. This saving would amount to more than 2 mDKK/year for the case system. The improvements provided by the proposed method are a step towards making non-convex optimization more reliable, which will allow for more complex models describing energy units and the relations between them in complex energy systems.

The authors suggest several points of improvement, namely running the optimizations of each time window in parallel to decrease computation and increasing the number of runs to further improve the consistency of the method. Other improvements should focus on creating heuristic rule sets to dynamically grow the search tree instead of searching a predefined set of actions. This is expected to increase the efficiency of the tree search considerably. Further research should focus on comparing the proposed method for non-convex optimization to methods involving convex approximation of the optimization problem.

Nomenclature

α_1 First order coefficient in pump work equation

- α_2 Second order coefficient in pump work equation
 α_3 Third order coefficient in pump work equation
 Φ_f Thermal power loss from return path of transmission line
 Φ_r Thermal power loss from return path of transmission line
 \dot{D}_H Sum of heat demand at consumer and distribution
 F Fuel consumption
 \dot{m}_v Outgoing mass flow rate at combined heat and power plant
 \dot{m}_x Mass flow rate to/from thermal storage
 \dot{m}_f Forward mass flow rate in transmission line
 P_E Electricity production at combined heat and power plant
 P_H Thermal power production at combined heat and power plant
 \dot{Q}_{flow1} Thermal energy flow from node to the thermal storage
 \dot{Q}_{flow2} Thermal energy flow from node to the transmission line
 $Q_{s,loss}$ Thermal heat loss from thermal storage
 $\dot{Q}_{s,pipe}$ Thermal energy flow in/out of the thermal storage
 \dot{W}_p Pump work
 η_E Electrical efficiency of combined heat and power plant in condensing mode
 $\pi(s)$ Policy state s
 a action
 A_s Area of heat conducting surface in thermal storage, which varies with contained mass
 c_v Constant relating heat and electricity production at a constant fuel consumption
 C_{p1} Constant pressure specific heat of water with unit $\frac{MJ}{ton^{\circ}C}$.
 C_{p2} Constants pressure specific heat of water with unit $\frac{MWh}{ton^{\circ}C}$
 EP Electricity price
 i Subscript, hour
 $M(s)$ Action set of state s
 m_s Contained mass in thermal storage
 MC Marginal cost of fuel consumption
 $Q_i(s, a)$ Expected cost of action a in state s
 Q_s Energy content of thermal storage
 s State
 T_a Ambient temperature used to calculate storage heat loss
 T_s Soil temperature used to calculate transmission heat loss
 T_v Temperature of outgoing water flow at combined heat and power plant
 T_x Temperature of water flow in pipe connecting to thermal
 T_{f1} Forward temperature in transmission line from node 1
 T_{f2} Temperature of district heating water at forward transmission line outlet
 T_{r1} Temperature of district heating water at return transmission line inlet
 T_{r2} Temperature of district heating water at return transmission line outlet
 $u_t(s, a)$ Exploration bias of action a in state s
 $U_{par}L$ Thermal power loss coefficient of transmission line
 W_L Length of the sliding time window
 W_S Stride of the sliding time windows

x Binary variable describing whether the thermal storage is charging or discharging
 z Binary variable describing the mode of operation of the combined heat and power plant

Abbreviations

LP: Linear programming; MILP: Mixed integer linear programming; MINLP: Mixed integer non-linear programming; NLP: Non-linear programming; MCTS: Monte Carlo Tree Search; TDKK: Thousand danish kroner; mDKK: Million danish kroner; BB: Branch and bound; TES: Thermal energy storage; UCT: Upper confidence applied to trees-algorithm; CHP: Combined heat and power

Acknowledgements

The authors want to acknowledge Fjernvarme Fyn for providing data to the project.

About this supplement

This article has been published as part of Energy Informatics Volume 4, Supplement 2 2021: Proceedings of the Energy Informatics Academy Conference Asia 2021. The full contents of the supplement are available at <https://energyinformatics.springeropen.com/articles/supplements/volume-4-supplement-2>.

Authors' contributions

JB developed the MCTS method. Based partly on the bachelor thesis of JB and LKM, which was under the supervision of CV and KF, JB and LKM developed the first draft of this work and CV, KF, HRS and MJ reviewed the work. All authors read and approved the final manuscript.

Funding

Not applicable

Availability of data and materials

NA.

Declarations

Ethics approval and consent to participate

NA.

Consent for publication

NA.

Competing interests

The authors declare that they have no competing interests.

Published: 24 September 2021

References

- APMonitor. APMonitor Optimization Suite, 2020a. <https://apmonitor.com/>, Accessed 3/02/2021
- APMonitor. APMonitor Documentation, 2020b. <https://apmonitor.com/wiki/index.php/Main/OptionApmSolver>, Accessed 3/02/2021
- Arcuri P, Florio G, Fragiaco P (2007) A mixed integer programming model for optimal design of trigeneration in a hospital complex. *Energy (Oxford)* 32(8):1430–1447
- Bindlish R (2016) Power scheduling and real-time optimization of industrial cogeneration plants. *Comp Chem Eng* 87:257–266. <https://doi.org/10.1016/j.compchemeng.2015.12.023>
- Boukouvava F, Misener R, Floudas CA (2016) Global optimization advances in mixed-integer non-linear programming, minlp, and constrained derivative-free optimization, cdf. *Eur J Oper Res* 252(3):701–727. <https://doi.org/10.1016/j.ejor.2015.12.018>
- Deng N, Cai R, Gao Y, Zhou Z, He G, Liu D, Zhang A (2017) A minlp model of optimal scheduling for a district heating and cooling system: a case study of an energy station in Tianjin. *Energy (Oxford)* 141:1750–1763. <https://doi.org/10.1016/j.energy.2017.10.130>
- Elsido C, Bischi A, Silva P, Martelli E (2017a) Two-stage minlp algorithm for the optimal synthesis and design of networks of chp units. *Energy* 121:403–426
- Elsido C, Bischi A, Silva P, Martelli E (2017b) Two-stage minlp algorithm for the optimal synthesis and design of networks of chp units. *Energy (Oxford)* 121:403–426. <https://doi.org/10.1016/j.energy.2017.01.014>
- Gopalakrishnan H, Kosanovic D (2015) Operational planning of combined heat and power plants through genetic algorithms for mixed 0–1 nonlinear programming. *Comput Oper Res* 56(C):51–67. <https://doi.org/10.1016/j.cor.2014.11.001>
- John D, Hedengren, Reza Asgharzadeh Shishavan, Kody M. Powell, and Thomas F. Edgar. Nonlinear modeling, estimation and predictive control in apmonitor. *Comput Chem Eng*, 70:133–148, 2014
- Kocsis L, Szepesvári C (2006) Bandit based monte-carlo planning, 17th European Conference on Machine Learning, pp 282–293
- Lésko M, Bujalski W, Futyma K (2018) Operational optimization in district heating systems with the use of thermal energy storage. *Energy (Oxford)* 165:902–915. <https://doi.org/10.1016/j.energy.2018.09.141>
- Lozano MA, Carvalho M, Serra LM (2009) Operational strategy and marginal costs in simple trigeneration systems. *Energy (Oxford)* 34(11):2001–2008

- Luo Y, Xigang Y, Yongjian L (2007) An improved pso algorithm for solving non-convex nlp/minlp problems with equality constraints. *Comput Chem Eng* 31(3):153–162
- Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T-t, Leach M, Kavukcuoglu K, Graepel T, Hassabis D, Silver D, Huang A (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529:484–489
- Makkonen S, Lahdelma R (2006) Non-convex power plant modelling in energy optimisation. *Eur J Oper Res* 171(3):1113–1126. <https://doi.org/10.1016/j.ejor.2005.01.020>
- Ramanujan R, Selman B (2011) Trade-offs in sampling-based adversarial planning. Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling, Freiburg
- Rong A, Lahdelma R (2005) An efficient linear programming model and optimization algorithm for trigeneration. *Appl Energy* 82(1):40–63. <https://doi.org/10.1016/j.apenergy.2004.07.013>
- Rong A, Lahdelma R (2007) An efficient envelope-based branch and bound algorithm for non-convex combined heat and power production planning. *Eur J Oper Res* 183(1):412–431. <https://doi.org/10.1016/j.ejor.2006.09.072>
- Savola T, Tveit T-M, Fogelholm C-J (2007) A minlp model including the pressure levels and multi-periods for chp process optimisation. *Appl Therm Eng* 27(11):1857–1867. <https://doi.org/10.1016/j.applthermaleng.2007.01.002>
- Soares J, Lobo C, Silva M, Morais H, Vale Z (2015) Relaxation of non-convex problem as an initial solution of meta-heuristics for energy resource management, IEEE Power & Energy Society General Meeting, vol 2015, Denver, pp 1–5
- Söderman J, Pettersson F (2006) Structural and operational optimisation of distributed energy systems. *Appl Therm Eng* 26(13):1400–1408. <https://doi.org/10.1016/j.applthermaleng.2005.05.034>
- The Danish Energy Agency and Energinet. Technology data, generation of electricity and district heating. 2019. https://ens.dk/sites/ens.dk/files/Analyser/technology_data_catalogue_for_el_and_dh.pdf, Accessed 3/02/2021
- Tveit T-M, Savola T, Gebremedhin A, Fogelholm C-J (2009) Multi-period minlp model for optimising operation and structural changes to chp plants in district heating networks with long-term thermal storage. *Energy Convers Manag* 50(3):639–647. <https://doi.org/10.1016/j.enconman.2008.10.010>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
