Chinese Journal of Chemical Engineering 69 (2024) 63–71

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

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

Full Length Article

Combining reinforcement learning with mathematical programming: An approach for optimal design of heat exchanger networks

Hui Tan [1, #], Xiaodong Hong [1, 2, #], Zuwei Liao [1, *], Jingyuan Sun [1], Yao Yang [1], Jingdai Wang [1], Yongrong Yang [1]

[1] State Key Laboratory of Chemical Engineering, College of Chemical and Biological Engineering, Zhejiang University, Hangzhou 310027, China
[2] ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou 311215, China

ARTICLE INFO

Article history:
Received 11 November 2023
Received in revised form
14 December 2023
Accepted 15 December 2023
Available online 29 December 2023

Keywords:
Heat exchanger network
Reinforcement learning
Mathematical programming
Process design

ABSTRACT

Heat integration is important for energy-saving in the process industry. It is linked to the persistently challenging task of optimal design of heat exchanger networks (HEN). Due to the inherent highly non-convex nonlinear and combinatorial nature of the HEN problem, it is not easy to find solutions of high quality for large-scale problems. The reinforcement learning (RL) method, which learns strategies through ongoing exploration and exploitation, reveals advantages in such area. However, due to the complexity of the HEN design problem, the RL method for HEN should be dedicated and designed. A hybrid strategy combining RL with mathematical programming is proposed to take better advantage of both methods. An insightful state representation of the HEN structure as well as a customized reward function is introduced. A *Q*-learning algorithm is applied to update the HEN structure using the ϵ-greedy strategy. Better results are obtained from three literature cases of different scales.

1. Introduction

Heat integration is an important means of reducing the energy consumption of chemical production processes, which is mainly achieved by designing heat exchanger networks (HEN) between process streams and utilities. The HEN problem was first proposed and defined by Masso and Rudd [1]. Since then, advancements have been achieved in this field, resulting in the proliferation of various methods with distinct applicability. These methods can be classified into sequential methods, based on sequential objectives and simultaneous methods that optimize multiple objectives simultaneously.

Sequential methods typically involve breaking down the HEN objective into multiple subobjectives, such as area, consumption of utility, units of heat exchangers, *etc*. The aim is to progressively minimize these objectives. Well-known sequential methods include the pinch design method proposed by Linnhoff and Flower [2] and several other works [3–5] that propose insight-based heuristics to target the surface area of heat exchangers, energy consumption, as well as capital cost. However, they are relying on engineers to design HENs manually. Other sequential methods use mathematical programming models. The transportation model proposed by Cerda *et al.* [6,7] and the transshipment model proposed by Papoulias and Grossmann [8] are typical sequential methods. Since sequential methods may lead to suboptimal solutions [9], subsequent researches shifted to simultaneous methods. They are formulated as mixed-integer nonlinear programming models. It should be noted that superstructures are frequently used in modeling HENs. The most widely used HEN superstructures are the generalized superstructure model proposed by Floudas *et al.* [10] and the stage-wise superstructure (SWS) model proposed by Yee and Grossmann [11]. The two kinds of superstructures have been extended by many researchers to various applications [12–15].

The algorithms for HEN optimization can be classified into two categories: deterministic algorithms and stochastic algorithms. Global optimality could be ensured by deterministic algorithms. Branch-and-bound [16,17], outer-approximation [18], as well as variable boundary contraction [19,20] are common techniques in deterministic algorithms for HEN optimization. Recently, Chang *et al.* [21–23] proposed a HEN structure enumeration strategy to achieve optimal HEN solution. However, due to the nonconvex and

* Corresponding author.
E-mail address: liaozw@zju.edu.cn (Z. Liao).
# These authors contributed equally to this work.

https://doi.org/10.1016/j.cjche.2023.12.005
1004-9541/

nonlinear nature of the problem, and the proved NP-hardness [24] property, these algorithms face challenges in handling large-scale problems. On the other hand, stochastic algorithms are promising in handling large-scale problems, although global optimality cannot be ensured. Stochastic evolutionary strategies include genetic algorithm [25], simulated annealing algorithm [26], particle swarm optimization algorithm [27], differential evolution algorithm [28], as well as some bionic algorithms imitating the behavior of organisms [29,30]. It should be noted that the aforementioned stochastic algorithms have not yet fully learned from the iterative process, while this is the direction of stochastic algorithms. The reinforcement learning (RL) method is good at this. RL enables learning how to maximize cumulative rewards through iteration between state and action; It is an efficient "trial-and-error" approach. It has been applied to the areas of production scheduling [31], process control [32], process design, and optimization [33,34]. The current RL applications to process design [33,34] are searching for small-scale flowsheets without HENs, where both discrete and continuous variables are included in the state representation. Although HEN is a part of the flowsheet, its structure combinatorial possibility is far greater than that of the other parts of the flowsheet. Consequently, the RL method for HEN should be dedicated and designed to gain an edge over the current algorithms for HEN, but there are no researches on this topic yet. We aim to put our insight of HEN into the RL-based algorithm design, including state representation, action adjustment, and reward evaluation. In addition, a hybrid strategy combining RL with mathematical programming is proposed to take advantage of both methods.

2. Problem Statement

The synthesis problem of HEN is defined as follows. Given are a set of hot (H)/cold (C) streams and a set of hot/cold utilities. The inlet and outlet temperatures, heat capacity flow rates, and heat transfer coefficients of each stream and utility are specified, as well as the cost factors. It is desired to obtain a HEN to minimize the total annual cost (TAC) while meeting the heat load requirements and approach temperature.

We adopt the well-known SWS model (shown in Fig. 1) [11] to present the HEN. By dividing the streams into several stages, the SWS model provides parallel, series, and hybrid HEN structures. This model is popular because it is simple and maintains structure variety. There may be loops in the HEN structure, and the HEN structure could be evolved by adding or reducing loops. We only consider HENs without loops in the SWS model. This is also known as minimal HEN structures, based on which the nonminimal structures can be evolved. The detailed model of SWS is shown in the Supplementary Material.

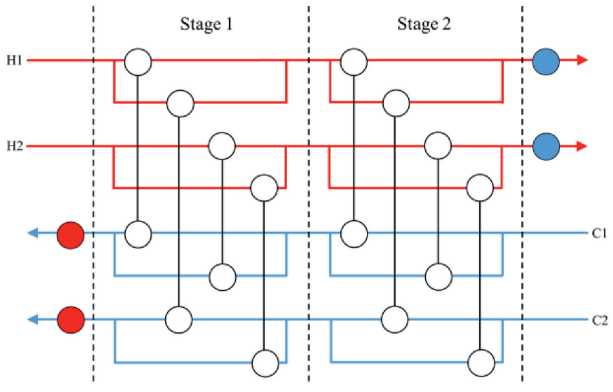

Fig. 1. The stage-wise superstructure (SWS) of heat exchanger network.

3. Methodology

RL is a machine-learning method that excels in optimizing decision-making through continuous trial and error, where an agent interacts with an environment, receiving rewards based on its actions, and aims to maximize the cumulative rewards over time. The strategy of combining RL with mathematical programming is shown in Fig. 2, which is composed of three blocks: state initialization, agent and environment. The three blocks are connected *via* three parameters: s, a and r, respectively, denoting the state, action, and reward of the RL. The state will be changed by implementing an action, and a reward can be calculated for an action derived from a state.

As shown in Fig. 2, the initial state s_0 is generated by the state initialization block and is sent to the other two blocks. When an action a_{it} is sent to the environment block, the current state can be updated. Then, the new state s_{it} will be translated to its corresponding HEN structure, and the TAC of the structure will be minimized *via* the nonlinear programming (NLP) model. A reward r_{it} can be calculated according to the obtained minimum TAC. The updated reward r_{it} and new state s_{it} will be transferred to the agent block, where a new action will be generated. The agent is the decision maker that selects the optimal action according to the current state. Since the action is only related to the current state and reward, the RL environment belongs to the Markov decision process.

3.1. State initialization

We need to generate a feasible HEN structure for the initial state s_0. Both pinch method and mathematical methods can generate feasible structures. A simple way of mathematical method is to solve the minimal SWS model without objective (MSWO, detailed in Supplementary Material). The resulting HEN structure serves as the initial state for both the agent and the environment in the RL-based approach.

3.2. State representation

One of the challenges of applying the RL methods to the HEN synthesis problem is the representation of HEN, which is also referred to as "state." The HEN can be represented by discrete variables describing its structure and continuous variables indicating the parameters of each heat exchanger (heat load, approach

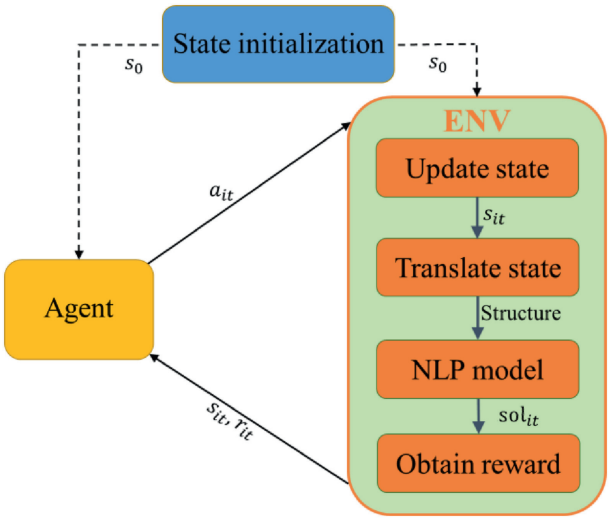

Fig. 2. The strategy based on reinforcement learning for optimal HEN design.

Table 1
Illustrative case

Stream	T_{in}/°C	T_{out}/°C	h/kW·m^{-2}·°C^{-1}	F_{cp}/kW·°C^{-1}
H1	270	160	0.5	18
H2	220	60	0.5	22
C1	50	210	0.5	20
C2	160	210	0.5	50
HU	250	250	1	
CU	15	20	1	
HE cost/USD·a^{-1}	4000 + 500 $A^{0.83}$			
Utility cost/USD·a^{-1}	200 Q_{HU} + 20 Q_{CU}			
EMAT	10 °C			

temperature, and flow rate of streams). Since the continuous variables make the state space wide and complex, because we have an efficient method to optimize those continuous variables when the discrete variables are fixed, only discrete variables describing HEN structures are involved in the state representation.

Let us consider the structure representation. According to the SWS model, there may exist a heat exchanger for a hot stream i and a cold stream j at a stage k. The existence of such a heat exchanger can be denoted by binary variables $x_{i,j,k} \in \{0,1\}$. The total number of the combinatorial space is calculated as $2^{N_i \times N_j \times N_k + N_i + N_j}$, where N_i, N_j, and N_k are the numbers of hot streams, cold streams, and stages, respectively. This space is too sparsely dispersed causing difficulties for the exploration of RL. Since a heat exchange match only includes one heat exchanger for the case of minimum HEN, we propose an alternative compact representation. The integer variable $x_{i,j} \in \{0, 1, \ldots, N_k\}$ is proposed to present the stage number of the heat exchanger between the hot stream i and cold stream j in the SWS model. When $x_{i,j}$ equals zero, the heat exchanger does not exist. Since heaters/coolers are only present at the outlet of cold/hot streams, their values can be represented by 1 and 0, respectively. Then the total number of the combinatorial space is calculated as $(N_K + 1)^{N_i \times N_j} \times 2^{N_i + N_j}$, which is smaller than $2^{N_i \times N_j \times N_k + N_i + N_j}$ when N_k is larger than 1. Let us take a simple case to illustrate the proposed representation. The data of the case are shown in Table 1, where 3 hot streams (including hot utility) and 3 cold streams (including cold utility) are shown. We can find 8 ($3 \times 3 - 1$) potential matches in this case. If we specify the k value to 2, then the total number of potential states is 1296, whereas if we use the traditional representation, the number should be 4096. Thus, the structure space is greatly reduced. Solving MSWO, we can obtain the initial structure, as shown in Fig. 3(a), where heat exchangers H1–C2, H2–C1, HU-C2, CU-H1, and CU-H2 are active, and they are all at the first stage. If we arrange all the 8 possible matches in the following order: H1–C1, H1–C2, H2–C1, H2–C2, HU-C1, HU-C2, CU-H1, and CU-H2, then the value of $x_{i,j}$ can be represented as a one-dimensional state array [0, 1, 1, 0, 0, 1, 1, 1]. As it is a feasible solution, the number of active heat exchangers (stage number greater than 0) equals to the specified minimum value. We call those kinds of states as end-point states. Otherwise, if a state with the number of active heat exchangers is less than the specified minimum number, then we call it a transition state.

The proposed state representation may generate repetitive HEN structures. For example, Fig. 3(a) and (b) are the same structures, but they are in different states. To eliminate such duplications, we calculate the minimum stage number of each heat exchanger in a state. The calculation algorithm is presented in Supplementary Material. If the stage number of a heat exchanger is not identical to its minimum value, we set the heat exchanger as inactive.

3.3. State access and reward calculation

The RL environment is named Structure Evolution in this paper. Following an action taken by the agent, the environment responds by providing the agent with a new state. This new state is translated to its corresponding HEN structure by translating $x_{i,j}$ to $x_{i,j,k}$. Then with these fixed $x_{i,j,k}$, the traditional SWS model is shifted to the minimal network optimization model (MNOM):

$$\begin{aligned}\text{Objective} = {} & C_{CU}\sum_{i\in H} q_{CU,i} + C_{CU}\sum_{j\in C} q_{HU,j} + C_{unit}\sum_{i\in H}\sum_{j\in C}\sum_{k\in K} x_{i,j,k} \\ & + C_a A_{i,j,k}^{\beta} \\ & C_{unit}\sum_{i\in H} x_{CU,i} + C_a A_{i,CU}^{\beta} + C_{unit}\sum_{j\in C} x_{HU,j} + C_a A_{HU,j}^{\beta} \end{aligned} \quad (1)$$

s.t.
Eq. (S1)–(S18) in Supplementary Material.

$$\left\{x_{i,j,k} = 1, x_{HU,j} = 1, x_{CU,i} = 1\right\} \ \forall (i,j,k) \in \text{ACT} \quad (2)$$

where ACT is the set of active heat exchangers in the new state. (A-1) to (A-12) are the formulas in the Supplementary Material. The HEN structure is fixed through Eq. (2). The minimum TAC of this structure can be obtained by solving MNOM, which is an NLP model. Considering the complexity of the objective function (Eq. (1)), we used the golden section algorithm proposed by Chang *et al.* [21] to optimize the TAC, based on the unimodal characteristic of the TAC as a function of the consumption of hot utility in the minimal network. Since the

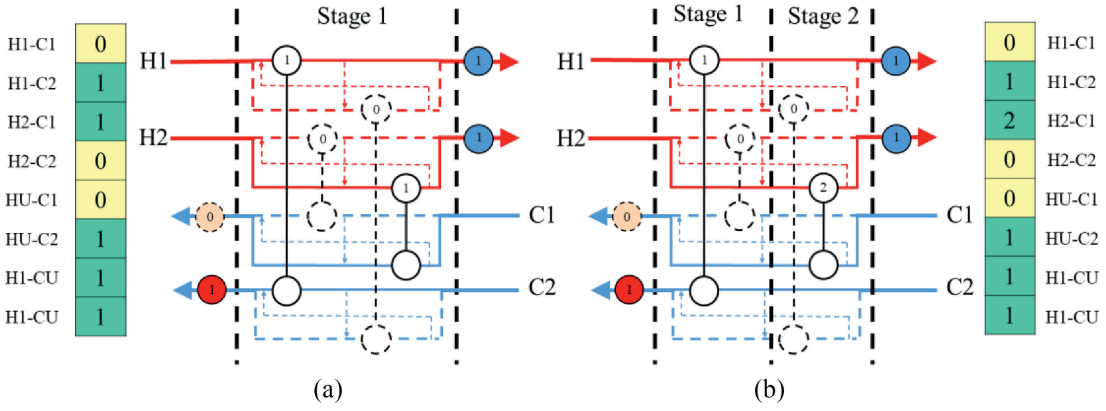

Fig. 3. Duplicate HEN structures for different stages.

transition state is infeasible, there is no need to run the MNOM model for this case. The obtained TAC will be sent back to the environment block to calculate the reward. If the new state is approved to be infeasible by the MNOM, the reward will be set to −1. When the new state is feasible, a reward will be calculated according to the optimal TAC of the new state, as shown by Eq. (3).

$$\text{Reward} = \begin{cases} -1, & s' \text{ is infeasible} \\ \exp\left(1 - \dfrac{\text{TAC}'}{\text{TAC}_b}\right), & s' \text{ is feasible} \end{cases} \tag{3}$$

where s' is the new end-point state, TAC_b is the optimal TAC of the current optimal state s_b, and TAC' is the optimal TAC of the new state s'. Note that if the TAC' is smaller than TAC_b, then both TAC_b and s_b will be updated to be TAC' and s', respectively. This reward function integrates exponential item and inverse proportionality item, and thus a positive reward is ensured. Furthermore, the reward of exponential growth will speed-up the searching procedure. To ensure the feasible state will provide a positive reward to the agent, we designed a reward function that integrates exponential and inverse proportionality. This design guarantees that as the TAC decreases, the agent receives greater incentives.

If the new end-point state s' is infeasible during the iteration, this new state will be discarded, and the current optimal state s_b will be sent to the agent for a new iteration. It means that the agent will determine a new action based on the current optimal state. This approach enables faster convergence to the optimal solution.

3.4. Action selection

To constrain the space of candidate actions, we specify that only one element in the one-dimensional state array is added or subtracted by 1 in each action. Let $l \in L$ denote the elements of the one-dimensional state array, then this action a can be denoted by $(l, \pm 1)$. For end-point states, the candidate elements are only among those activated ones. For transition states, we choose the activated elements with the probability of ζ while choosing the inactivated elements with the probability of $1-\zeta$. Then, the ϵ-greedy algorithm is applied to guide the selection of actions for both activated and inactivated elements. Specifically, the agent randomly selects the actions in the search space with a relatively small probability ϵ, facilitating efficient exploration of unknown strategies and actions. Furthermore, it selects the action with the largest expected reward (Q-value) in the current state with probability $1-\epsilon$. This deliberate balance between exploration and exploitation ensures the effective utilization of known strategies and actions to attain enhanced rewards.

The action a taken in state s is evaluated using the Q-value, $Q(s, a)$, that represents the expected reward from taking action a in state s. $Q(s, a)$ is defined by Bellman equation [35]:

$$Q(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} \left[\max_{a'} Q(s', a') \right] \tag{4}$$

where $p(s'|s,a)$ represents the state transfer function of the environment, the probability distribution of the state transfer after performing action a in state s. γ is the discount factor. Note that $r(s,a)$ and $p(s'|s,a)$ in Eq. (4) are difficult to get as the complete environment model in real scenarios is difficult to obtain. Alternatively, we use Q-learning [36], a sample-based method, to approximate the Bellman equation and Eq. (4) is substituted by

$$Q(s,a) = (1-\alpha) Q(s,a) + \alpha \left[r(s,a) + \gamma \max_{a'} Q(s', a') \right] \tag{5}$$

where α is the learning rate. Note that if an action is not allowed, then there is no corresponding Q-value.

3.5. Illustrative case

Let us take the illustrative case in Table 1 to illustrate the iterative procedure between agent and environment. The initial state s_0 of this case is obtained in the previous section by solving the MSWO model. There may be tens of thousands of iterations to find the optimal solution. For ease of representation, we only select six actions of two iterations both starting from s_0 to illustrate the iterative procedure, as shown in Fig. 4. The previously identified one-dimensional state array of 8 elements is denoted by a row of 8 squares in the figure, and

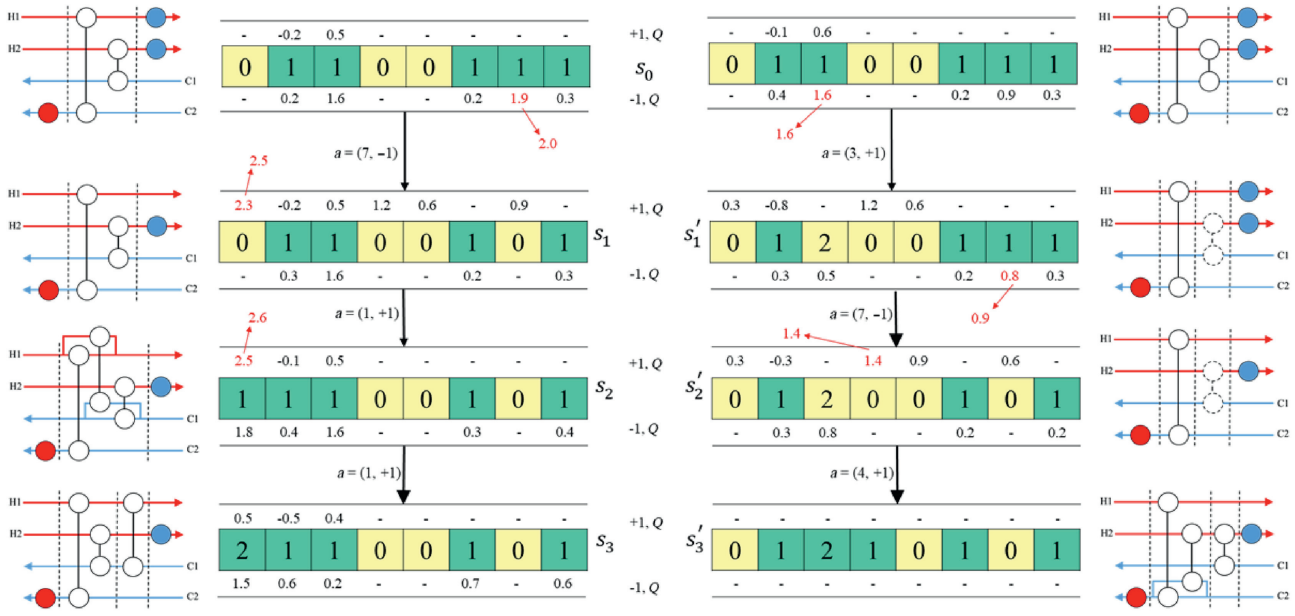

Fig. 4. Illustrative state evolution in two scenarios.

their corresponding HENs are shown beside the row of squares. The green and yellow colors stand for the activated and inactivated states of the heat exchangers. The current Q-values corresponding to the candidate actions are also provided below (minus 1 action) or above (plus 1 action) the squares. Note that the current Q-values are determined by the previous iterations. Probability parameters ϵ and ζ are set at 0.1 and 0.7, respectively, for this case and the following case studies. Take the left scenario as a case. The agent decides to choose an action according to the current Q-values of possible actions for the state s_0. The candidate action with the largest Q-value (1.9) is chosen, and the corresponding action is (7, −1), which means the stage number of the 7th heat exchanger (H1-CU) is changed from 1 to 0. Thus the 7th heat exchanger is turned from active to inactive. The state is updated to s_1. Then, $Q(s_0,(7,-1))$ can be updated to 2.0 according to the reward presented in Table 2. Since s_1 is a transition state, the reward is set to zero. Running the action-selection rule, we obtain the second action: (1, 1), and the state is updated to s_2. Then the reward of s_2 and the Q-values of s_1 can be calculated sequentially. Similarly, the third action can be decided as still (1, 1), and thus the state is updated to s_3. From the right scenario, state s'_1 is a duplicated state of state s_0. This is because the stage number of the third heat exchanger is larger than the minimum number. As a result, it is set to be inactive, although the number is not zero, but at state s'_3, it is back to active, for the minimum number is changed to 2. Applying the MNOM model to state s'_3, it is found to be infeasible. Thus, this scenario of iteration is ended, and the next state is updated to the current best state.

3.6. Algorithm

The pseudocode describing the environment and agent is shown in Algorithm 1.

Algorithm1. Pseudocode for the learning cycle that considers the environment and the agent

Obtain the initial state s_0 from MSWO
Pass the state s_0 to the agent and the environment
Agent

Get the action list$_{it}$ based on the current state s_{it}
if exploit:
  a_{it} = arg_max (Q[s_{it}, a]), ∀a∈action list$_{it}$
else if explore
  a_{it} = Random (action list$_{it}$)
Pass a_{it} to the environment (SE)
r_{it+1}, s_{it+1} ← SE
$Q(s_{it},a_{it}) \leftarrow (1-\alpha)Q(s_{it},a_{it}) + \alpha\left[r(s_{it},a_{it}) + \gamma \max_{a_{it+1}} Q(s_{it+1},a_{it+1})\right]$

SE

a_{it+1} ← **Agent**
s_{it+1} ← update state s_{it} by action a_{it+1}
s_{it+1} ← adjust elements that make the structure repeat
if s_{it+1} is endpoint state
  res_{it+1} ← running MNOM
  r_{it+1} = Reward (res_{k+1})
else if s_{it+1} is a transition state
  r_{it+1} = 0
Pass r_{it+1}, s_{it+1} to **Agent**

4. Case study

Three cases of different sizes are selected from the published literatures to illustrate the effectiveness of the developed RL-based method. The stopping criterion for the program is to stop after reaching a predetermined number of RL iterations or upon the elapse of a specific time duration. Our program was implemented in Python (version 3.8.16), and the model was constructed within the Pyomo framework (version 6.5.0). The solver used for the model was Gurobi for academics (version 10.0.1). The following case studies were conducted on a work station equipped with AMD EPYC 7702 2 GHz processors and 192 GB of RAM.

Table 2
TAC and rewards for the illustrative states in Fig. 4

	s_1	s_2	s_3	s'_1	s'_2	s'_3
TAC_b	–	361007.3	361007.3	–	–	
TAC	–	382678.5	360038.3	–	–	–
Reward	0	0.94	1.02	0	0	–

Table 3
Data of case 1

Stream	T_{in}/°C	T_{out}/°C	h/kW·m^{-2}·°C^{-1}	F_{cp}/kW·°C^{-1}
H1	226.85	46.85	2	6
H2	206.85	106.85	2	4
H3	186.85	86.85	2	6
H4	106.85	360	2	20
H5	106.85	46.85	2	12
C1	16.85	386.85	2	18
HU	700	426.85	2	
CU	26.85	46.85	1	
HE cost/USD·a^{-1}	5500 + 1200 $A^{0.6}$			
Utility cost/USD·a^{-1}	140 Q_{HU} + 10 Q_{CU}			
EMAT	10			

4.1. Case 1

Case 1 was proposed by Escobar and Grossmann [37] and was also applied to a minimal network study by Chang *et al.* [21]. This case consists of five hot streams, one cold stream, one hot utility, and one cold utility, as shown in Table 3. The optimal TAC of this case changes with the number of RL iterations. An iteration represents that RL reaches an infeasible state from the initial state, and the following meanings are the same. The optimal TAC of the initial structure obtained by MSWO is 696747.92 USD·a^{-1}. The optimal solution of 649365.88 USD·a^{-1} is obtained after 1175 iterations in 2.1s, and its structure is shown in Fig. 5. In the following 8825 iterations within 11.78 s, no structures with smaller TAC are found.

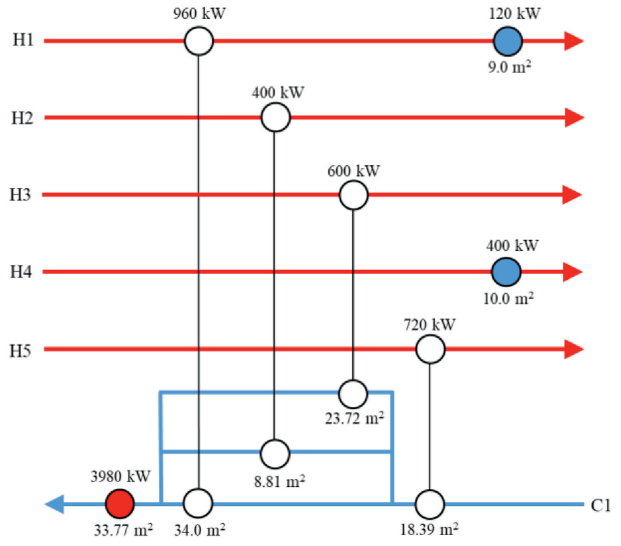

Fig. 5. The optimal structure of Case 1.

Table 4
Results comparison of Case 1

Case 1	Unit	Q_{HU}/kW	A/m^2	TAC/USD·a^{-1}
Chang *et al.* [21]	6	4580.0	109.4	724506.4
Initial result	7	4313.3	125.23	696747.9
Final result	7	3980.0	137.68	649365.9

Table 5
Data of Case 2

Stream	T_{in}/°C	T_{out}/°C	h/kW·m^{-2}·°C^{-1}	F_{cp}/kW·°C^{-1}
H1	840	40	1.5	4.9894
H2	76	45	1.5	4.684
H3	50	40	1.5	0.772
H4	180	77	1.5	0.6097
H5	180	179	0.8	292.7
H6	90	45	1.5	3.066
C1	24	25	0.8	329.8
C2	25	70	1.5	0.5383
C3	35	122	1.5	3.727
C4	90	180	1.5	0.6097
C5	180	181	0.8	2581.1
HU	230	230	1.5	
CU	20	40	0.8	
HE cost/USD·a^{-1}	$9094 + 485\ A^{0.81}$			
Utility cost/USD·a^{-1}	$110\ Q_{HU} + 15\ Q_{CU}$			
EMAT	10 °C			

Table 6
Results comparison of Case 2

Case 2	Unit	Q_{HU}/kW	A/m^2	TAC/USD·a^{-1}
Castillo *et al.* [38]	11	0	–	141554
Silva *et al.* [39]	11	0	66.15	140142
Pavão *et al.* [40]	11	0	64.69	140068
Pavão *et al.* [27]	11	0	65.11	139438
Chang *et al.* [21]	11	0	65.09	139398.1
Initial result	11	0	67.4	140256.8
Final result	11	0	65.08	139388.7

This is because the optimal structure within the realm of the minimal networks has been achieved. The comparison results with the literature are shown in Table 4. Chang *et al.* [21] obtained the optimal TAC of 724506.4 USD·a^{-1}, where the number of heat exchanger units is set to 6. However, according to the definition of the minimal network, the maximum number of heat exchanger units allowed for this case is 7. A better structure, achieving a TAC that is 10% lower than the result obtained by Chang *et al.* [21] is obtained *via* the proposed method.

4.2. Case 2

Case 2 is derived from a real case of a nitric acid plant in Barrancabermeja, Colombia, investigated and studied by Castillo *et al.* [38]. This case was then adapted by Silva *et al.* [39] to make it

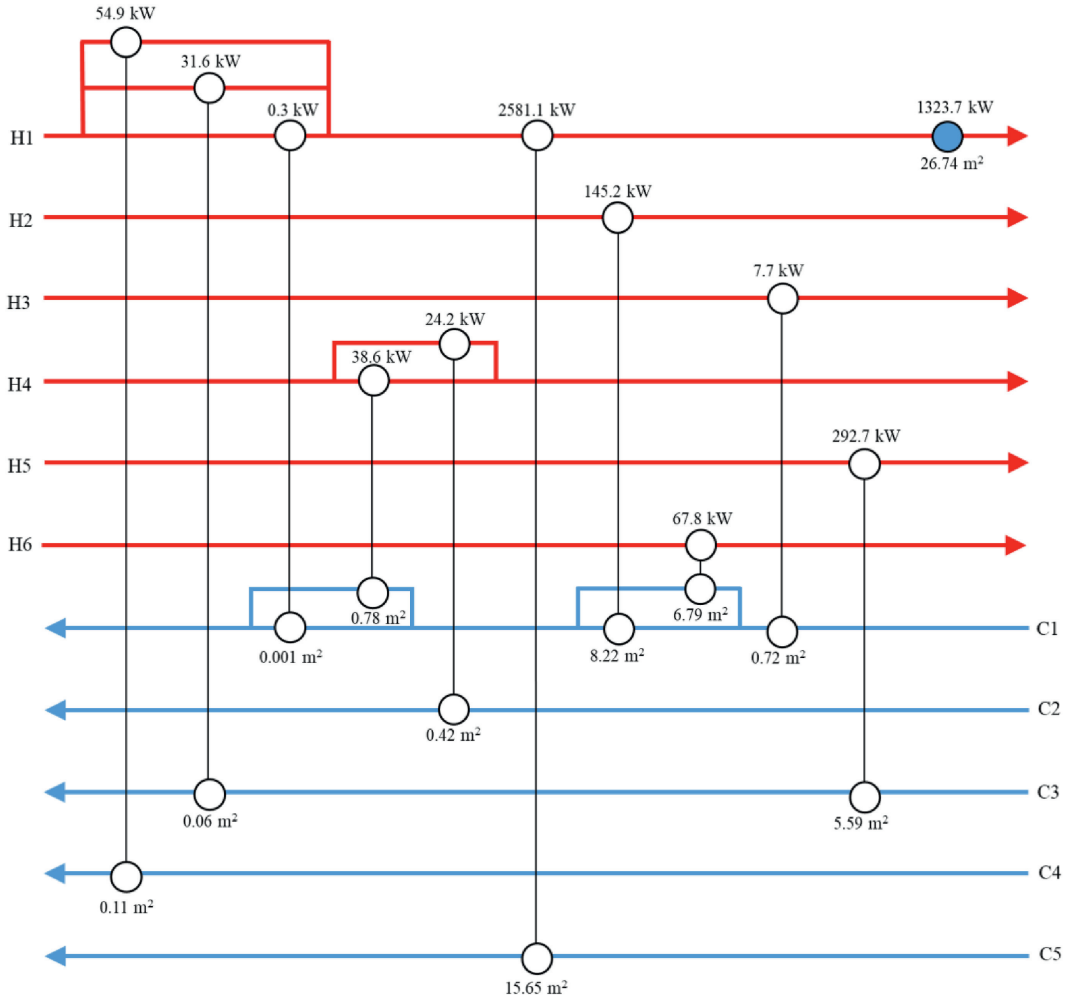

Fig. 6. The optimal structure of Case 2.

Table 7
Data of Case 3

Stream	T_{in}/°C	T_{out}/°C	h/kW·m^{-2}·°C^{-1}	F_{cp}/kW·°C^{-1}
H1	576	437	0.06	23.1
H2	599	399	0.6	15.22
H3	530	382	0.06	15.15
H4	449	237	0.06	14.76
H5	368	177	0.06	10.7
H6	121	114	1	149.6
H7	202	185	1	258.2
H8	185	113	1	8.38
H9	140	120	1	59.89
H10	69	66	1	165.79
H11	120	68	1	8.74
H12	67	35	1	7.62
H13	1034.5	576	0.06	21.3
C1	123	343	0.06	10.61
C2	20	156	1.2	6.65
C3	156	157	2	3291
C4	20	182	1.2	26.63
C5	182	318	1.2	31.19
C6	318	320	2	4011.83
C7	322	923.78	0.06	17.6
HU	927	927	5	
CU	9	17	1	
HE cost/USD·a^{-1}	$4000 + 500\ A^{0.83}$			
Utility cost/USD·a^{-1}	$250\ Q_{HU} + 25\ Q_{CU}$			
EMAT	8			

applicable to academic research. The case data are shown in Table 5, which include six hot streams, five cold streams, one hot utility, and one cold utility. The optimal TAC of the initial structure obtained by MSWO is 140256.79 USD·a^{-1}. Then after 15 iterations within 0.03s, a better structure is found, and the TAC is reduced to 140199.44 USD·a^{-1}. The TAC stopped updating until 47820 iterations, and then multiple updates were performed. The optimal solution 139388.69 USD·a^{-1} is found after 61117 iterations within 232.4s. The optimal structure is shown in Fig. 6, which includes 11 units with a total heat exchanger area of 65.08 m^2.

The obtained results are compared with the literatures, as shown in Table 6. Silva *et al.* [39] used PSO (particle swarm optimization) algorithm to obtain an optimal result of 139777 USD·a^{-1}. Pavão *et al.* [40] corrected the TAC of the optimal result reported by Silva *et al.* [39] to 140142 USD·a^{-1}. Pavão *et al.* [40] used a parallel approach combined with heuristic algorithms such as GA (genetic algorithm) and PSO algorithm to obtain a result of 140068 USD·a^{-1} and then used the rocket fireworks algorithm [27] to further improve the result to 139438 USD·a^{-1}. Chang *et al.* [21] proposed a global optimization algorithm tailored to the minimal networks. The optimal result of Chang *et al.* [21] is 139398.1 USD·a^{-1}. Although Chang *et al.* [21] used a global optimization algorithm, they have applied termination criteria (*e.g.*, CPU time or relative gap between lower and upper bounds) for large-scale cases. The developed RL-based method obtains a better result of 139388.7 USD·a^{-1} within 232.4 s, which shows that the algorithm's performance in exploring the searching space.

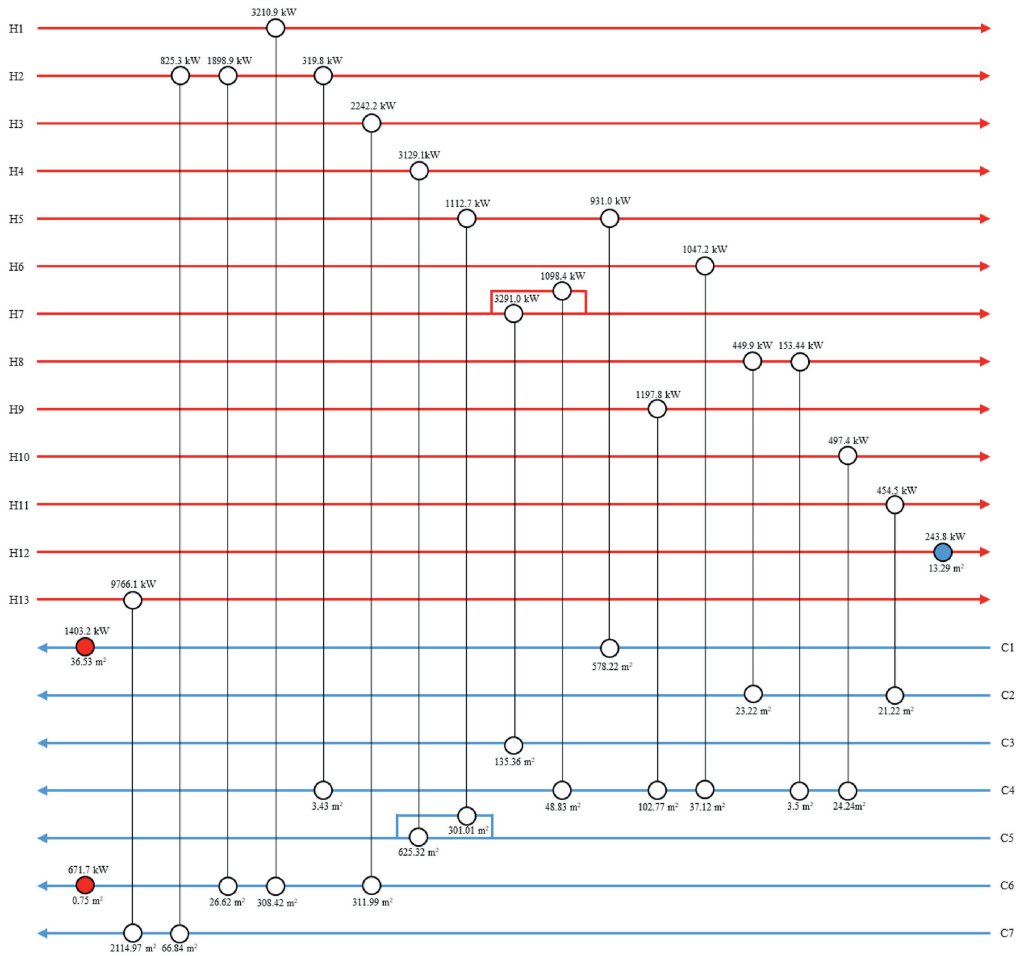

Fig. 7. The optimal structure of Case 3.

Table 8
Results comparison of Case 3

Case 3	Unit	Q_{HU}/kW	A/m^2	TAC/USD·a^{-1}
Escobar and Trierweiler [9]	21	1938.0	551.08	1537086
Pavão *et al.* [40]	21	1938.0	5389.01	1516482
Xiao and Gui [42]	23	2198.3	5077.79	1549795
Zhang and Cui [30]	22	1831.1	5110	1418981
Chang *et al.* [21]	21	1831.1	5190.9	1427966
Initial result	20	2328.4	6452.97	1659172
Final result	20	2074.9	4770.36	1405803

4.3. Case 3

Case 3 is a large-scale problem consisting of 13 hot streams and 7 cold streams, adapted from the work of Escobar and Trierweiler [9] and Soršak and Kravanja [41]. The case data are shown in Table 7. Case 3 is much more difficult to solve because there are more local optimal solutions for large-scale problems. The optimal TAC of the initial structure obtained by MSWO is 1659172.01 USD·a^{-1}. The TAC was then updated several times and reduced to 1659172.01 USD·a^{-1} at 1270 iterations, taking a total of 16.14s. Subsequently, the TAC was reduced to 1444237.41 USD·a^{-1} at 143544 iterations, and the optimal solution 1405803 USD·a^{-1} was obtained at 1209218 iterations, which takes a total time of 7608.34 s. The optimal result is shown in Fig. 7**,** which contains a total of 20 units.

The obtained results of this study are compared with five literature results, as shown in Table 8. Escobar and Trierweiler [9] obtained an optimal solution of 1461006.2 USD·a^{-1}, using a discrete and continuous optimizer based on their proposed initial-point generation strategy for the SWS model. However, Pavão *et al.* [40] pointed out that the heat exchanger area was miscalculated by Escobar and Trierweiler [9] and then corrected the result to 1537086 USD·a^{-1}. Besides, Pavão *et al.* [40] obtained an optimal TAC of 1516482 USD·a^{-1} using a parallel heuristic algorithm. Xiao and Gui [42] reported an optimal TAC of 1153146 USD·a^{-1} using a new random walk algorithm. Zhang and Cui [30] found that the reported structure *via* Xiao and Gui [42] has a TAC of 1549795 USD·a^{-1}, which was then optimized to 1418981 USD·a^{-1} using the cuckoo search algorithm. Finally, Chang *et al.* [21] reported an optimal TAC of 1427966 USD·a^{-1}. The TAC of the five literatures is all slightly higher than the optimal TAC obtained in our work, which is 1405803 USD·a^{-1}. It is shown that the developed RL-based method also has better optimization performance on large-scale cases.

5. Conclusions

This work uses the RL method for the optimal design of HENs. A hybrid strategy combining RL with mathematical programming is developed, where RL is applied for HEN structure optimization and mathematical programming is applied for minimizing the TAC of the fixed HEN structure. A novel compact HEN structure representation based on the SWS model is proposed. We applied RL to optimize the HEN structures by utilizing customized strategies for structure adjustment and evolution. The reward function integrates the exponential item and inverse proportionality item, which ensure a positive reward and will speed-up the searching procedure. These strategies were combined with ϵ-greedy and *Q*-learning techniques. Three cases of different sizes are used to verify the effectiveness of the proposed method, which demonstrates better results.

Declaration of 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 financial support provided by the Project of National Natural Science Foundation of China (U22A20415, 21978256, 22308314) and “Pioneer” and “Leading Goose” Research & Development Program of Zhejiang (2022C01SA442617) are gratefully acknowledged. Besides, the authors would like to thank artificial intelligence + High Performance Computing Center of ZJU-ICI.

Supplementary Material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cjche.2023.12.005.

Nomenclature

$A_{i,CU}$	area of heat cold utilities, m^2
$A_{HU,j}$	area of heat hot utilities, m^2
$A_{i,j,k}$	area of heat exchangers, m^2
a	the action of reinforcement learning
C_a	area cost coefficient of heat exchangers, USD·a^{-1}·m^{-2}
C_{CU}	the cost coefficient of cold utilities, USD·a^1·kW^{-1}
C_{HU}	the cost coefficient of hot utilities, USD·a^{-1}·kW^{-1}
F_{CP}	heat capacity flow rate, kW·°C^{-1}
h	heat transfer coefficient, kW·m^{-2}·°C^{-1}
i	the serial number of hot streams
it	iteration of reinforcement learning
j	the serial number of cold streams
k	the serial number of stages
L	one-dimensional state array
l	element of one-dimensional state array
$\mathrm{Lmtd}_{i,CU}$	logarithmic mean temperature difference of cold utilities, °C
$\mathrm{Lmtd}_{HU,j}$	Logarithmic mean temperature difference of hot utilities, °C
$\mathrm{Lmtd}_{i,j,k}$	Logarithmic mean temperature difference of heat exchangers, °C
N_i	number of hot streams
N_j	number of cold streams
N_k	number of stages
Q_{HU}	consumption of hot utilities, kW
Q_{CU}	consumption of cold utilities, kW
T_{in}	inlet temperature of streams, °C
T_{out}	outlet temperature of streams, °C
$q_{i,j,k}$	load of heat exchangers, kW
$q_{CU,i}$	load of cold utilities, kW
$q_{HU,i}$	load of hot utilities, kW
r	the reward of reinforcement learning
s	the state of reinforcement learning
sol	solution of NLP model
$x_{i,j,k}$	binary variable for heat exchangers
$x_{i,j}$	stage number of heat exchangers
$x_{CU,i}$	binary variable for cold utilities
$x_{HU,j}$	binary variable for hot utilities
α	learning rate
β	exponent of area cost
γ	discount factor
ε	greedy strategy probability
ζ	probability for selecting heat exchanger

References

[1] A.H. Masso, D.F. Rudd, The synthesis of system designs. II. Heuristic structuring, *AlChE. J.* 15 (1) (1969) 10–17.

[2] B. Linnhoff, J.R. Flower, Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks, *AlChE. J.* 24 (4) (1978) 633–642.
[3] B. Linnhoff, S. Ahmad, Cost optimum heat exchanger networks—1. Minimum energy and capital using simple models for capital cost, *Comput. Chem. Eng.* 14 (7) (1990) 729–750.
[4] S. Ahmad, B. Linnhoff, R. Smith, Cost optimum heat exchanger networks—2. Targets and design for detailed capital cost models, *Comput. Chem. Eng.* 14 (7) (1990) 751–767.
[5] R. Colberg, Area, Cost and Resilience Targets for Heat Exchanger Networks, Dissertation (Ph.D.), California Institute of Technology, Pasadena, USA, 1989.
[6] J. Cerda, A.W. Westerberg, D. Mason, B. Linnhoff, Minimum utility usage in heat exchanger network synthesis A transportation problem, *Chem. Eng. Sci.* 38 (3) (1983) 373–387.
[7] J. Cerda, A.W. Westerburg, Synthesizing heat exchanger networks having restricted stream/stream matches using transportation problem formulations, *Chem. Eng. Sci.* 38 (10) (1983) 1723–1740.
[8] S.A. Papoulias, I.E. Grossmann, A structural optimization approach in process synthesis—III, *Comput. Chem. Eng.* 7 (6) (1983) 723–734.
[9] M. Escobar, J.O. Trierweiler, Optimal heat exchanger network synthesis: A case study comparison, *Appl. Therm. Eng.* 51 (1–2) (2013) 801–826.
[10] C.A. Floudas, A.R. Ciric, I.E. Grossmann, Automatic synthesis of optimum heat exchanger network configurations, *AlChE. J.* 32 (2) (1986) 276–290.
[11] T.F. Yee, I.E. Grossmann, Simultaneous optimization models for heat integration—II. Heat exchanger network synthesis, *Comput. Chem. Eng.* 14 (10) (1990) 1165–1184.
[12] S.Y. Kim, M. Bagajewicz, Global optimization of heat exchanger networks using a new generalized superstructure, *Chem. Eng. Sci.* 147 (2016) 30–46.
[13] K.M. Björk, T. Westerlund, Global optimization of heat exchanger network synthesis problems with and without the isothermal mixing assumption, *Comput. Chem. Eng.* 26 (11) (2002) 1581–1593.
[14] K.F. Huang, E.M. Al-mutairi, I.A. Karimi, Heat exchanger network synthesis using a stagewise superstructure with non-isothermal mixing, *Chem. Eng. Sci.* 73 (2012) 30–43.
[15] P. Jongsuwat, U. Suriyapraphadilok, M. Bagajewicz, New heat exchanger network design model, *Chem. Eng. Trans.* 39 (2014) 121–126.
[16] J.M. Zamora, I.E. Grossmann, A comprehensive global optimization approach for the synthesis of heat exchanger networks with no stream splits, *Comput. Chem. Eng.* 21 (1997) S65–S70.
[17] C.S. Adjiman, I.P. Androulakis, C.A. Floudas, Global optimization of MINLP problems in process synthesis and design, *Comput. Chem. Eng.* 21 (1997) S445–S450.
[18] M.L. Bergamini, I. Grossmann, N. Scenna, P. Aguirre, An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms, *Comput. Chem. Eng.* 32 (3) (2008) 477–493.
[19] S.Y. Kim, P. Jongsuwat, U. Suriyapraphadilok, M. Bagajewicz, Global optimization of heat exchanger networks. Part 1: Stages/substages superstructure, *Ind. Eng. Chem. Res.* 56 (2017) 5944–5957.
[20] D.C. Faria, S.Y. Kim, M.J. Bagajewicz, Global optimization of the stage-wise superstructure model for heat exchanger networks, *Ind. Eng. Chem. Res.* 54 (5) (2015) 1595–1604.
[21] C.L. Chang, A. Peccini, Y.F. Wang, A.L.H. Costa, M.J. Bagajewicz, Globally optimal synthesis of heat exchanger networks. Part I: Minimal networks, *AlChE. J.* 66 (7) (2020) 16267.
[22] C.L. Chang, Z.W. Liao, A.L.H. Costa, M.J. Bagajewicz, Globally optimal synthesis of heat exchanger networks. Part II: Non-minimal networks, *AlChE. J.* 66 (7) (2020) 16264.
[23] C.L. Chang, Z.W. Liao, A.L.H. Costa, M.J. Bagajewicz, Globally optimal synthesis of heat exchanger networks. Part III: Non-isothermal mixing in minimal and non-minimal networks, *AlChE. J.* 67 (11) (2021) 17393.
[24] K.C. Furman, N.V. Sahinidis, Computational complexity of heat exchanger network synthesis, *Comput. Chem. Eng.* 25 (9–10) (2001) 1371–1390.
[25] M.A.S.S. Ravagnani, A.P. Silva, P.A. Arroyo, A.A. Constantino, Heat exchanger network synthesis and optimisation using genetic algorithm, *Appl. Therm. Eng.* 25 (7) (2005) 1003–1017.
[26] F.Y. Peng, G.M. Cui, Efficient simultaneous synthesis for heat exchanger network with simulated annealing algorithm, *Appl. Therm. Eng.* 78 (2015) 136–149.
[27] L.V. Pavão, C.B.B. Costa, M.A. da Silva Sá Ravagnani, L. Jiménez, Large-scale heat exchanger networks synthesis using simulated annealing and the novel rocket fireworks optimization, *AlChE. J.* 63 (5) (2017) 1582–1601.
[28] K.M. Yerramsetty, C.V.S. Murty, Synthesis of cost-optimal heat exchanger networks using differential evolution, *Comput. Chem. Eng.* 32 (8) (2008) 1861–1876.
[29] M. Dorigo, G. Di Caro, Ant colony optimization: A new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE, Washington, DC, USA, 2002, pp. 1470–1477.
[30] H.L. Zhang, G.M. Cui, Optimal heat exchanger network synthesis based on improved cuckoo search *via* Lévy flights, *Chem. Eng. Res. Des.* 134 (2018) 62–79.
[31] C.D. Hubbs, C. Li, N.V. Sahinidis, I.E. Grossmann, J.M. Wassick, A deep reinforcement learning approach for chemical production scheduling, *Comput. Chem. Eng.* 141 (2020) 106982.
[32] J.C. Hoskins, D.M. Himmelblau, Process control *via* artificial neural networks and reinforcement learning, *Comput. Chem. Eng.* 16 (4) (1992) 241–251.
[33] A. Khan, A. Lapkin, Searching for optimal process routes: A reinforcement learning approach, *Comput. Chem. Eng.* 141 (2020) 107027.
[34] L. Stops, R. Leenhouts, Q.H. Gao, A.M. Schweidtmann, Flowsheet generation through hierarchical reinforcement learning and graph neural networks, *AlChE. J.* 69 (1) (2023) 17938.
[35] R. Bellman, Dynamic programming, *Science* 153 (3731) (1966) 34–37.
[36] C.J.C.H. Watkins, P. Dayan, Technical note: Q-learning, *Mach. Learn.* 8 (3) (1992) 279–292.
[37] M. Escobar, I.E. Grossmann, Mixed-Integer Nonlinear Programming Models for Optimal Simultaneous Synthesis of Heat Exchangers Network, 2010. www.minlp.org/library/problem/index.php?i=93.
[38] E. Castillo, L. Acevedo, A. Reverberi, Cleaner production of nitric acid by heat transfer optimization: A case study, *Chem. Biochem. Eng. Q.* 12 (2018) 157–165.
[39] A.P. Silva, M.A.S.S. Ravagnani, E.C. Biscaia, J.A. Caballero, Optimal heat exchanger network synthesis usingparticle swarm optimization, *Optim. Eng.* 11 (3) (2010) 459–470.
[40] L.V. Pavão, C.B.B. Costa, M.A. da Silva Sá Ravagnani, Automated heat exchanger network synthesis by using hybrid natural algorithms and parallel processing, *Comput. Chem. Eng.* 94 (2016) 370–386.
[41] A. Soršak, Z. Kravanja, Simultaneous MINLP synthesis of heat exchanger networks comprising different exchanger types, *Comput. Aided Chem. Eng.* 9 (2001) 1095–1100.
[42] Y. Xiao, G.M. Cui, A novel Random Walk algorithm with Compulsive Evolution for heat exchanger network synthesis, *Appl. Therm. Eng.* 115 (2017) 1118–1127.