Expert Systems With Applications 258 (2024) 125192

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods

Sándor Szénási [a,b,*], Gábor Légrádi [a]

[a] John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary
[b] Faculty of Economics and Informatics, J. Selye University, Komarno, Slovakia

ARTICLE INFO

Keywords:
Machine learning
Metaheuristics
Local search
Hybrid method
Optimization

ABSTRACT

Machine learning-based methods have emerged as competitors to traditional metaheuristic-based solutions in many areas. Besides investigating their effectiveness, it raises the question of whether these methods can be combined. This paper presents a systematic literature review based on P.R.I.S.M.A. methodology to provide a state-of-the-art overview of machine learning-assisted metaheuristics, focusing on local search algorithms such as Hill Climbing, Tabu Search, and Simulated Annealing. The review is based on a comprehensive evaluation of 48 related articles. These studies illustrate the most common applications of hybrid methods in various fields, including physical simulations and scheduling problems. This paper demonstrates commonly used assembly options, such as metamodeling and machine learning aided initialization, along with some novel ideas like early stopping and cooling control based on neural networks. The evaluation of the results reveals several potential machine learning methods, such as Deep Neural Networks, Hopfield Networks, and Self-Organizing Maps, to assist the metaheuristics. Different training methods for these approaches, including online vs offline training and sources of training data, are also reviewed. Most papers address real-world problems, but there are several intriguing ideas for improving local searches in general.

Contents

* Corresponding author at: John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary.
E-mail addresses: szenasi.sandor@nik.uni-obuda.hu, szenasis@ujs.sk (S. Szénási), legradi.gabor@nik.uni-obuda.hu (G. Légrádi).

https://doi.org/10.1016/j.eswa.2024.125192
Received 8 December 2023; Received in revised form 24 June 2024; Accepted 22 August 2024
Available online 30 August 2024

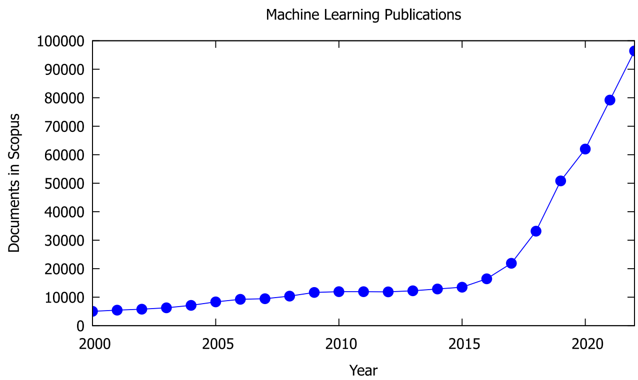

Fig. 1. Number of publications about machine learning.

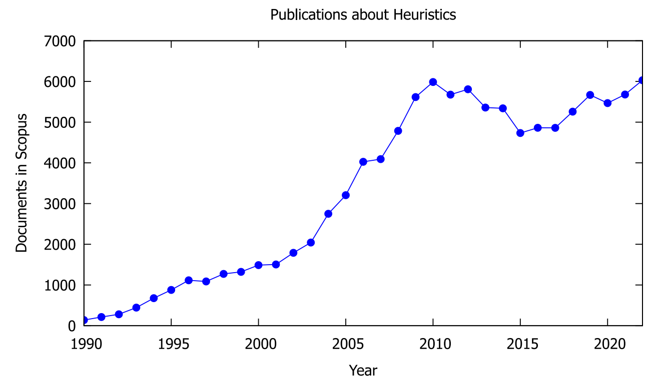

Fig. 2. Number of publications about metaheuristics.

1. Introduction

We are currently experiencing a renaissance in Artificial Intelligence (AI) research, driven by significant advancements in the technology industry. Which has made it possible to put long-known theoretical methods into real-world practice. Recent developments in data storage, fast-access memory, and almost unlimited cloud-based computing capacity have enabled the practical application of long-established theoretical methods. These technological advancements have revitalized many research areas, leading to dynamic developments. New innovations are being announced by manufacturers, novel results are being presented by researchers, and AI-based systems are increasingly being adopted across industrial and research fields.

This trend is illustrated in Fig. 1, which shows the number of publications listed in Scopus between 2000 and 2022 that include the terms "machine learning", "deep learning" or "neural" in the title. Following an initial slow growth, an exponential increase has been observed since 2015.

However, this paradigm shift has stalled the development of some methods refined decades before the new AI explosion. In many practical applications, traditional methods have been replaced by AI-based solutions, such as the shift from traditional image processing algorithms to AI-based approaches. This raises the question of whether a similar change has occurred in the field of metaheuristics (MH). Are the new AI-based solutions capable of refining existing developments in this field?

Looking at the number of publications (Fig. 2), a steady growth since the 1990s slowed around 2010, followed by a brief decline. It is assumed that many researchers have turned away from this field. The figure shows the number of articles published in Scopus with the terms "hill climbing", "simulated annealing", "tabu search", "genetic algorithm", "ant colony optimization", and "particle swarm optimization" in the title. This list is not exhaustive but sufficient for an illustration.

The aim of this paper is to explore how emerging Machine Learning (ML) techniques affect existing Local Search (LS) algorithms. Are there ways to combine these methods and build more effective solutions?

Several review articles on this topic have been published in recent years. However, their focus differs significantly from the aim of this paper. A recent review by Rajwar, Deep, and Das (2023) provides a state-of-the-art classification of MHs but does not specifically discuss ML-assisted hybrids. The paper by Karimi-Mamaghan, Mohammadi, Meyer, Karimi-Mamaghan, and Talbi (2022) presents hybrid implementations in remarkable depth; but focus specifically on combinatorial problems, which is narrower than the area we are investigating (mainly focusing on the methods rather than the problem to be solved). Conversely, it is worth highlighting the work of Talbi (2021), who reviewed and classified nearly 300 papers in the field of hybrid solutions. The article examines the capabilities at three levels (problem-level, low-level ML support, and high-level ML support), of which the middle one is the most relevant to the topic of this research. The article provides an excellent taxonomy for the field, but it cannot dive deeply (even at the algorithm level) into the innovative variations. In contrast to the previous ones, da Costa Oliveira, Britto, and Gusmão (2023) have already done a systematic review of hybrid algorithms. Since they have studied the whole spectrum of MHs, they have processed a considerable number of papers, so it was only possible to present them in general terms, and the paper mainly deals with their possible classification.

While these review articles and preliminary studies reveal many ML-MH hybrid applications, most can be classified into well-known use cases. These hybrids often loosely relate the two methods, such as when a pre-trained neural network accelerates fitness computation or estimates an initial state for the MH. Due to the broad focus of these reviews, the main groups are identified and detailed. However, there are individual innovative ideas representing a tighter integration between the two methods at a low level. These are fewer mainstream applications, therefore remaining hidden from the scope of these review articles.

This motivates our paper, where we explore specific research initiatives to encourage further exploration in this direction. We investigate and present each solution in depth, even at the algorithm level if necessary, categorizing them accordingly. Such a detailed analysis is possible only by narrowing the focus to the investigation considerably, in this case, to the subject of local searches. We evaluated hybrid versions of the following methods: general Local Search, Hill Climbing (HC), Tabu Search (TS), and Simulated Annealing (SA).

LS methods are relatively simple, providing a convenient way to study the potential of hybrid algorithms. They involve some non-trivial operations (e.g., the memory of TS, the temperature function of SA, etc.), offering opportunities to exploit the advantages of complex ML extensions. Their simplicity allows for extensive experiments to validate the advantages and disadvantages of various hybrid implementations. The insights gained from the study of LS methods will be valuable when investigating MHs with higher resource requirements. This paper seeks to answer the following questions:

- Are machine learning methods already used in local search algorithms?

- Are there novel theoretical results, or are the applications mainly practical?
- In which research areas are these hybrid solutions emerging?
- How can the two different approaches be assembled?
- What local search methods can be assisted by machine learning models?
- Which machine learning models can be integrated into these hybrid solutions?
- Is the machine learning component trained online or offline?
- What are the possibilities for producing training data?

A systematic review begins with the proper collection of papers on the given topic. According to the P.R.I.S.M.A. guidelines (Page et al., 2021), relevant papers were collected from Scopus and W.O.S. databases.

The structure of the paper corresponds to the steps of the P.R.I.S.M.A. method: 1. identification (Section 2.1), 2. screening (Section 2.2), 3. inclusion (Section 3).

Section 4 presents the evaluation of the selected papers, and the final section draws conclusions from the research.

2. Methodology

2.1. Identification step

In the identification step, search keywords were determined to find relevant papers. The main goal was to find papers about ML-assisted LS methods. For this, the following keyword rules were constructed:

- The title of the paper must contain any of the words "machine learning", "deep learning", or "neural" to ensure the relation of machine learning methods.
- The title of the paper must also contain any of the following words "local search", "hill climbing", "tabu search", or "simulated annealing" to ensure that the paper is related to local search metaheuristics.
- The publication year must be 2000 or after.

In the Scopus database, the following advanced query was used: TITLE (("machine learning" OR "deep learning" OR "neural") AND ("local search" OR "hill climbing" OR "tabu search" OR "simulated annealing")) AND PUBYEAR > 1999

The number of articles in Scopus meeting these criteria was 315.

In the WoS database, the following query was used: TI = (("machine learning" OR "deep learning" OR "neural") AND ("local search" OR "hill climbing" OR "tabu search" OR "simulated annealing")) AND PY = (2000–2023)

The number of articles in WoS fitting this filter was 230.

2.2. Screening step

As a preliminary step, duplications were removed from the results. This was easily handled based on the DOI, authors, title, and source. There were 220 papers included in both databases, resulting in 93 unique papers in Scopus and 10 in WoS. Two papers were added manually, bringing the final number of papers to 325.

The next step was obtaining the full texts from the following sources: directly from publishers or indirectly from Scopus, ScienceDirect, IEEE Xplore, Google Scholar, ResearchGate, or from one of the authors. 15 of the 325 articles could not be obtained from any source.

No automation tools were used for the screening process; therefore, all the abstracts were read by a human expert who classified the papers into three groups:

- The paper is in the scope of this research.
- The article is undoubtedly outside the scope.
- The paper needs further investigation.

Due to inadvertent errors, the whole process was run twice, and only articles always selected for the excludable category were excluded. At the end of this preprocessing, 47 of the 310 papers were removed from further processing.

All the remaining 263 articles were read carefully and evaluated based on the given methodology. Articles excluded from subsequent detailed analysis were grouped into the following categories:

- Reverse functionality: This paper focuses on MHs supported by ML and not the reverse; therefore, any research in which MH and ML were implemented sequentially in the reverse order was excluded.
- Neural network training: Because supervised training is highly computationally demanding, several attempts have been made to modify the backpropagation algorithm using the ideas of SA or TS; however, these are out of the scope of this paper.
- Neural network architecture optimization: Automatic configuration of neural networks using MH methods is widely used in practice; however, these can be considered as MH-assisted ML procedures.
- Sequential steps: Many solutions were excluded where MH and ML appear in more or less independent successive steps.
- Independent use of methods: Articles where MH and ML methods appear independently have also been excluded, including articles comparing their operation on a specific task.
- Technical reasons: Articles returned by the search terms by coincidence but unrelated to the present research were also excluded.

3. Presentation of selected papers

After the above steps, 48 articles remained on the list. This chapter briefly presents these papers, highlighting their target research areas, methods, and results.

3.1. Machine learning assisted metaheuristic initialization

Rajan, Mohan, and Manivannan (2003) presented a novel neural-based TS method to solve the short-term unit commitment problem of power systems. This requires the determination of the on/off states of generators to minimize the operating cost for a given time horizon. In the first phase, an ANN is used as a pre-processor. The input of the ANN is a particular day and its corresponding status; the output provides the ON-OFF statuses of the units. This is a quick but only approximate solution to the problem. Therefore, a further LS is necessary to fine-tune it. It is done by a TS algorithm, in which input is given by this ANN output. The results of the proposed model are far superior to those produced by conventional methods. The training set for the neural network is collected in previous load patterns and their corresponding commitment schedules. A case study shows that the proposed method is 20% faster and gives 5% more accurate results. C.C.A. Rajan (with various co-authors) has published multiple papers in the field. With a similar methodology, these articles focus on various practical problems: unit commitment problem for utility system (Rajan & Mohan, 2007) and unit commitment problem with cooling-banking constraints (Rajan, 2009). They also propose an ANN-SA hybrid for the same problem (Rajan, Mohan, & Manivannan, 2002).

El-Bouri, Balakrishnan, and Popplewell (2005) presented a paper solving the n-job, m-machine permutation flow shop problem to minimize the mean flowtime. Their method designs initial sequences to improve the performance of the LS. These are created from job rankings provided by an ANN. The network is trained using data from optimal sequences obtained from solved small-sized problem examples. Once trained, the ANN provides rankable measures that can be used to construct a sequence in which jobs are located as close as possible to the positions they would occupy in an optimal sequence. These results are further refined by a TS method. Tests demonstrate that the sequences recommended by the ANN are effective in guiding LS.

Yang (2006) presented a Job Shop Scheduling Problem (JSSP) solver using an ANN and LS hybrid approach. He built a flexible ANN that follows the rules of JSSP. It finds possible schedules for the problem, and the LS tries to improve these by looking at similar ones. Test results show that the new technique can find solutions with 10% better fitness.

Fadaei and Setayeshi (2008) presented a novel ANN and SA-based tool for loading pattern optimization for reactor cores. The authors employed a Continuous Hopfield Network (CHN) to approximate a semi-optimal loading pattern. The network output can be considered a local minimum; therefore, an SA algorithm is initiated to escape local minima and reach the global minimum. The outcome is near to the global optimal configuration, which concurs with the pattern suggested by the designer. The proposed algorithm found a 10% better solution than the already used method.

The paper of Jemai and Mellouli (2008) presented a neural-TS heuristic for the real-time vehicle routing problem. It is a variant of the dynamic routing problem where the requests are not known in advance but appear randomly; the objective is to respond and assign the requests as quickly as possible. The authors proposed a two-phase approach: a feed-forward ANN learns and predicts the routing decisions from previous instances followed by TS starting from the solution given by the ANN and improves it further. A mathematical model is used for the static version of the real-time vehicle routing problem to generate the training data for the ANN.

Lugon Jr, Silva Neto, and Santana (2009) presented a hybrid approach for the solution of gas-liquid adsorption inverse problems. This study combines an ANN, the SA method, and the Levenberg–Marquardt method (LM) to solve the inverse problem efficiently and accurately. The ANN is trained with the direct problem solutions and provides an initial estimate for the LM method, which is a gradient-based deterministic local optimizer. The SA method is applied to explore the neighborhood to avoid being trapped in a local minimum. If the SA method does not improve the solution obtained by the LM method, it indicates a high probability of reaching the global minimum. Using the ANN to calculate the derivatives in the initial steps of the LM method reduces the computational time significantly.

Mohebi and Sap (2009) presented an optimized hybrid Kohonen Self-Organizing Map (SOM) for ambiguity detection in cluster analysis using SA. A two-level approach is proposed for the data analysis: the data is processed by the SOM at the first level, and a rough set-based incremental clustering approach is applied to the output of the SOM at the second level. The uncertainty that comes from some clustering operations is minimized by a SA algorithm that has been adopted. The experimental results demonstrate that the proposed algorithm outperforms the existing crisp clustering algorithms.

Liu, Ye, Si, and Xu (2013) presented a SA/ANN hybrid method to determine the optimum initial operation pressure of a steam turbine. Initially, an ANN-based model was developed to regulate the sliding pressure characteristics of the turbine. Additionally, a biogeographic optimum algorithm based on SA was introduced. This algorithm combines the ability of the ANN to quickly identify the optimal solution and the ability of the SA algorithm to perform LSs. As a result, the search precision and convergence speed were significantly improved.

Bouhouch, Bennis, Loqman, and El Qadi (2018) introduced a CHN-based approach to solve the Constraint Satisfactory Problem. This problem generally consists of a finite set of variables, each with a finite domain of values and a set of constraints. The goal is to find a complete assignment of variables that satisfies all constraints. The proposed approach is based on the basic idea of using the Min-Conflict Algorithm (MCA) to improve the solution reached by CHN. Several tests were run to show that the proposed algorithm is faster and has better fitness than the existing methods. The rate of network success in providing a valid solution is up to 100%. The average fitness was 33.8% better, and the average runtime was 33.1% of the runtime of the traditional methods.

Vitali, Mele, Gambardella, and Montemanni (2021) present a novel method composed of two phases to solve the Traveling Salesman Problem (TSP). As a preliminary step, the ML method was trained by thousands of randomly generated instances. The first phase exploits the ability of ML to detect specific patterns to create an initial partial solution. According to the ML learned patterns, this solution comprises the edges most likely to be part of an optimal tour. Suitable approximated tours are provided by different ML methods (ANN, Support Vector Machine - SVM), which can, however, still be improved. These tours have some parts that could be improved since some crossing edges can get in them. The second phase uses heuristics to complete the solution. The use of SVM improves quality compared to the original method, and an improvement in speed of about 4× is also achieved.

The paper of Shao and Kim (2022) proposes an effective hybrid method based on a Deep Convolutional Neural Network (CNN) connected to an SVM classifier and iterative LS to solve JSSP with small-scale training samples and less time. The specific ANN is designed to generate a global path by treating JSSP as a series of sub-classification tasks. The global path is then refined by the LS to obtain a local optimum. The trained model is used to predict the priority of each machine for each suboperation for large-scale JSSP instances. The proposed method outperforms traditional methods: the average score is 1.2%–31.4% better than other frequently used methods.

Tian et al. (2023) presented a multi-objective optimization algorithm for an energy-efficient disassembly line balancing problem. The initial solution for disassembly is produced through K-means clustering, which expedites individual information exchange. Subsequently, innovative methodologies for the enhancement of disassembly sequences are formulated, incorporating a LS strategy to increase the precision of the algorithm. Comparative analysis with prevalent algorithms substantiates the superior performance of the proposed whale optimization algorithm, demonstrating an optimal equilibrium between solution quality and efficiency.

3.2. Machine learning based metamodeling

In this section, several similar approaches, called metamodeling, are presented. They are based on the idea that the costly fitness calculation is approximated by a simpler model. The direct effect is a reduction in runtime. However, it can indirectly lead to improved accuracy (the local search can do many more iterations in the same amount of time).

Su and Chang (2000) presented an ANN and SA-based approach for parameter design optimization. The methodology proposed consists of two distinct stages. First, an ANN is utilized to outline the correlation between input and output variables; this trained model is subsequently employed to predict the response for a given set of parameters. Following this, the SA algorithm is implemented using the ANN to give a fast prediction for the fitness value to identify the optimal response and its associated parameter configuration. The ANN is trained offline with a given set of sample input–output values. In the case of a numerical example, the accuracy of the novel method is twice that of the already-known best method.

Chen and Yang (2002) presented a manufacturing system design using a hybrid approach with ANN metamodeling and stochastic LS. A simulation model for the studied problem is first constructed by the authors. Sets of training and testing data are generated by running the simulation experiments. This method is commonly used in metamodeling when it is too expensive to produce sufficient training samples. Simulations can quickly generate large amounts of noise-free data, although their usefulness is more limited than real-world data. The ANN learns the relationships between the input variables and simulation responses. Once the relationships have been learned, the SA can try to locate the optimal combination of design variables using the metamodel as an objective function. The proposed methodology effectively and efficiently solves a complex manufacturing system design problem.

Rao, Thandaveswara, Murty Bhallamudi, and Srinivasulu (2003) introduced an optimal groundwater management system using SA and ANN. An SA algorithm that interacts with an existing SHARP simulator interface flow model searches for the optimal policy for the location and pumpages of cooperative wells. An ANN is employed to replace the simulator to reduce the computational cost. The ANN is trained with several input/output pairs generated by the simulator. This improvement significantly decreases the simulation time, and the fitness calculation time decreases to near zero. Rao has a similar publication (Rao, 2006) where the methods are the same, but the objective of the optimization is the efficient identification of unknown contaminant sources.

Maslov (2003) worked in the area of image registration to find an affine transformation minimizing the difference between images. A hybrid evolutionary algorithm enhanced with a LS phase is used, which leads to better chromosomes in the LS space. A response matrix for the entire image is built in the preprocessing stage to reduce the amount of computations. Different points are classified according to their response values by a SOM. The map is used as a lookup table to retrieve the appropriate values of the response. The novel approach significantly reduces the number of evaluations, constituting 34% computational cost reduction, without compromising solution quality.

The study of Hsu (2004) presents an integrated optimization approach based on ANNs, exponential desirability functions, and TS to resolve the multi-response parameter design problem using the traditional Taguchi method. First, a backpropagation ANN is trained to establish the functional correlation between input control factors and output responses. Training data is collected by specific experiments. This network is subsequently employed as a mathematical estimative model for the TS algorithm, accelerating the fitness calculation process. The experimental results indicate that the average defect rate has been reduced to approximately 1.0%, which is a significant improvement from the previous rate of over 15%.

The paper of Lahiri and Chakravorti (2005) deals with the optimization of stress distribution on and around electrode-spacer arrangements for ensuring economical and higher reliability of gas-insulated systems. The optimization is based on an SA algorithm, which is greatly enhanced by coupling it with a trained ANN calculating the cost function to evaluate the optimum values for the design parameters of the electrode-spacer arrangements. The training and test data were generated by electrostatic-field calculation using indirect boundary element formulation. The total time required for the execution is 4× less for the novel method.

Gao and Tian (2007) presented a path-planning algorithm for mobile robots based on an improved SA-ANN. The energy function of this algorithm consists of two terms: the obstacle collision penalty function (which is expressed using an ANN) and the path length. The algorithm aims to find a solution that minimizes the energy function, thus achieving a short and collision-free path for the mobile robot. The simulation results demonstrate that the novel method is 2.32× faster, and the average length of paths is 9.7% less.

Hajji, Fares, Glover, and Driss (2010) presented a water pump scheduling system using scatter search, TS, and ANN. As a prediction model, the ANN accelerates the search by avoiding evaluating the fitness function for a newly created reference point. The training of the ANN is based on analyzed historical and synthetic demands. The result of the novel optimization engine comes with an optimal water pumping plan that decreases energy costs by 23%.

Abbasi and Mahlooji (2012) designed a novel ANN- and SA-based improved response surface methodology. This method primarily encompasses two consecutive stages. The initial stage involves the approximation of the objective function by employing a first-order polynomial. Subsequently, the second stage entails the estimation of the same objective function, this time utilizing a second-order polynomial. In the present study, an ANN is employed to approximate the response surface, followed by the application of a SA algorithm to identify the optimal or near-optimal response. The proposed model has been evaluated against several mathematical benchmark functions, demonstrating superior performance.

Rathinam and Kannan (2014) presented a paper about optimizing the ferrite number of duplex stainless steel cladding using ANN and SA. Independent regression and ANN-based estimators are built to predict ferrite numbers based on various process parameters. The training of the ANN is done by using experimental data. As a consecutive step, an SA is launched to find the optimal parameters to gain the highest ferrite number. The ANN-based estimator is used to set the initial position for the SA and to define boundaries for the search. The integrated ANN-SA method is an effective technique to get the optimal process values.

Pan, Duque, Martins, and Debenest (2020) applied a neural fuzzy model combined with an SA algorithm to predict optimal conditions for polyethylene waste non-isothermal pyrolysis. In the initial phase, an adaptive neural fuzzy model was employed to forecast the conversions and pyrolysis rates of both virgin and waste polyethylene. Thermogravimetry experiments at different heating rates were conducted to provide the training data for the adaptive neural fuzzy model. Subsequently, the SA was utilized to optimize the optimal operating conditions across various temperature ranges.

Yadav, Tripathi, Asati, and Das (2020) proposed an ANN-based surrogate framework along with an SA approach to inversely estimate the optimum values of heat source control parameters in a heat treatment furnace. Firstly, an ANN-based surrogate is developed from the data generated by a forward radiative transfer equation solver. This metamodel is then coupled with the SA algorithm to solve the multi-variable optimization problem and obtain optimum values of heat source control parameters. The ANN accurately estimates the fluxes, with maximum errors less than 6% for such predictions. Furthermore, the ANN-based model is significantly faster than the radiate transfer equation solver by an order of 10^5s.

Huang, Li, He, Zhang, and Wong (2021) presented a novel ANN and SA-based method to optimize antenna design. The data set utilized for training the ANN is produced by a simulation software. The trained ANN is capable of swiftly predicting the electromagnetic parameters of the antenna; therefore, it is used as the cost function of the optimization. Subsequently, a SA is employed to optimize the geometric parameters of the antenna in order to achieve the desired value.

The work of Sheng, Sun, and Liu (2021) presented a hybrid NN and SA to explore the effect of reaction conditions on the yield of C4 olefins. A mathematical model is formulated to investigate the catalyst combination and reaction temperature in the chemical reaction. Subsequently, an ANN is architected and trained using the input/output pairs derived from the mathematical model. In the final stage, SA is employed to identify the optimal parameters, where the fitness function is determined by the trained ANN. The model demonstrates robust performance with goodness-of-fit tests.

Sen, Rai, Chakrabarty, Lahiri, and Dutta (2021) presented a method based on ANN and SA to optimize the phycoremediation of Cr(VI). In the initial phase, an ANN is trained by experimental data to model the non-linear complex relationship between chromium removal percentage with various input parameters. Following this, a SA algorithm is initiated, assuming arbitrary values for four input parameters within their high-low limit. These values are then passed to the trained ANN model to determine the percentage of chromium removal. The SA algorithm iteratively modifies the input parameters in accordance with its algorithm until the minimization of the objective function is achieved.

Shao et al. (2022) presented an optimization method using improved ANN and SA for soil sampling design for generating accurate soil maps. An improved ANN was first developed and then utilized to calculate the objective function of SA. The target variable of the optimization is soil organic matter content, which is critical in biological processes. The training and testing data for the ANN are from prior real-world samples. The proposed vs reference sampling strategy resulted in an improvement in accuracy of 12.5%.

Khosravinia and Kiani (2023) designed an ANN/SA hybrid to optimize ultrashort laser pulses for high-performance supercapacitor electrodes. An extensive dataset was synthesized to find the correlation between the parameters of laser fabrication and the electrochemical behavior performance. This was achieved through the application of an ANN. The fitness calculation phase of the SA process involves invoking the trained ANN to find out the value of a given candidate solution. The implementation of SA has been found to enhance the performance of the pseudocapacitor and can be effectively integrated into optimal design procedures.

Yokoi, Kato, Oshima, and Matsunaga (2023) presented an ANN-driven SA method to predict grain boundary structures in Si and Ge. To identify the minimum-energy configurations, the SA technique is applied, which relies on molecular dynamics simulations and utilizes the potentials derived from an ANN. The training dataset for the ANN is generated from some reference structures by creating multiple variations of each structure with random alterations in their cell dimensions and atomic positions and performing relaxation simulations to record the snapshots into the training database.

The work of Liao, Yu, Wang, Dong, and Yang (2023) is another example of metamodeling from the field of designing topological photonic crystals. Before the launch of MH, an ANN is trained to predict the band structure of crystals. This network is embedded into the SA algorithm to speed-up the fitness calculation. Their solution demonstrates the efficiency and flexibility of this hybrid method.

3.3. Machine learning in addition to metamodeling and initialization

The paper of Persson, Grimm, and Ng (2006) presented a new algorithm for enhancing the efficiency of simulation-based optimization using LS and ANN metamodels. Steepest ascent HC is used as the basis for the LS strategy. A metamodel for simulation optimization is employed by this algorithm, which differs from many other approaches that use them sequentially because it uses them by alternating between the metamodel and its underlying simulation model. This technique prevents the accumulation of minor inaccuracies in the metamodel that could push the search in a completely wrong direction. Indeed, the conventional heuristic steps set a generally acceptable state after a certain number of steps. Another difference is that the accuracy of the metamodel in the current region of the search space is improved by applying online learning. This is also a useful change as it allows the model to be better adapted to the specific problem space. In addition, this method reduces the time needed for prior training and the amount of training data required. A theoretical benchmark problem and a real-world manufacturing optimization problem are used to test the proposed algorithm, and its performance is compared to a standard HC strategy, showing promising results.

The work of Hakimi-Asiabar, Ghodsypour, and Kerachian (2009) is based on a multi-objective genetic LS, a hybridization of genetic algorithm and LS. In the proposed method, the hybrid algorithm is extended with a grid of neurons using the learning rule of SOM. SOM is a neural network capable of learning; in this case, it is trained by the Pareto dominance of solution candidates. The knowledge of historic Pareto fronts stored in the SOM helps determine new solutions for the GA. During the execution of the genetic algorithm, the SOM is trained by the chromosomes of the first Pareto front. As a next step, a SOM-assisted local Variable Neighborhood Search (VNS) is conducted to ensure convergency. The novel method outperforms the NSGA-II in terms of speed and accuracy.

The paper of Grzechca (2011) presents a new hybrid procedure for analog circuit fault clustering using SA and SOM. The main goal of the method is to find the best piece-wise linear excitation under the maximum diagnosability of the circuit. SA is used for optimization, where the goal function ranks how well an ANN performs in classifying the circuit of interest for each node in the optimization. The testing procedure is performed in the time domain, and the piece-wise linear excitation is optimized by SA. The input data for the neural network comes from the test points of the circuit. The SOM is applied to cluster all circuit states into possible separate groups. The hybrid approach shows good efficiency in a reasonable time.

In the paper of Fister et al. (2016), the ANN regression is applied as a LS heuristic within the Differential Evolution (DE) algorithm that tries to predict the best strategy to generate a better offspring from an ensemble of DE strategies. DE is a widely used heuristic, but it is hard to select the proper DE crossover/mutation strategy for a given problem; furthermore, the best strategy may vary during a given search process. The proposed DE variant introduces a LS step after generating trial vectors. During each LS process, a regression ANN is trained on the elements derived from a given chromosome using randomly selected DE strategies. Ideally, the trained ANN can predict the best solution from the set of trial solutions provided by the available DE strategies. Because this procedure is very computationally intensive, this LS is applied only with a given probability. The results of experiments conducted on a CEC 2014 test suite consisting of 30 benchmark functions have shown that the proposed hybrids substantially outperform their original predecessors. Moreover, the performance gap broadened when the dimensionality of the problem was increased. The median of the average improvement is 8.1%.

Bożejko, Gnatowski, Niżyński, and Wodecki (2016) presented a solution for the NP-hard cyclic JSSP of task scheduling. A TS algorithm using a neural mechanism to prevent looping was implemented to solve the problem. The novel idea is that the model works similarly to the biological mechanism of forgetting, which differs from the traditional Tabu list because a given move is not explicitly prohibited or allowed. A unique neural network architecture is responsible for this decision, where every neuron corresponds to a move that changes the execution order of two given operations. Based on the online training, this network can predict the best solution from a given neighborhood and work as a memory, taking on the role of the Tabu list. Computational experiments were conducted to show the statistically significant efficacy of the proposed TS method compared to the classical list of forbidden moves. Because the usual approach has a couple of problems: it requires a large amount of storage space to store the moves, and a lot of computing capacity to search here. A ML-based model can eliminate both problems simultaneously. The proposed method works better in the case of large problem sizes.

Hao and Yoshimura (2017) presented a novel method for many heuristic optimization algorithms. One of the most common methods to reduce execution time and improve solution optimality is to estimate the quality of a set of candidate solutions and examine only the most promising candidates in detail. This step can save time, but it needs a very accurate estimator. In contrast to the standard way, the authors proposed an online learning-based estimator that can fine-tune itself during the search process, improving estimation accuracy. A simple case study is also presented, where a local search-based heuristic with random start was used, and an online estimator considering the properties of the LS was proposed. The experiments showed that the accuracy of the online estimator is much higher than that of the static one and even higher than that of a general offline pre-trained learner. Although the online estimator needs additional training time, the heuristic algorithm still speeds up by 3.7× without sacrificing optimality.

van de Ven, Zhang, and Lee (2019) presented a novel solution for a capacity determination problem of shunting yards. Two hybrid ML/LS methods are proposed. The first method uses a Deep Graph CNN to predict the feasibility of finding a shunt plan with LS from a random initial solution. A classification model is built and trained using instances generated from a simulator, and it is shown that this approach can significantly reduce the planning time for determining the capacity of a given shunting yard. The second method explores the use of ML to guide LS in exploring promising areas during the search. The specific LS applies 11 consecutive search operators in every iteration. These order is predicted by using a similar Graph Neural Network (GNN). The training data is generated by the simulator containing graphs from initial, intermediate, and final solutions, extended by a label for the search operators that were applied to improve the solution in the next iteration. The comprehensive evaluation of the novel method shows that accurately predicting the evaluation order can lead to faster and more consistent improvement of solutions. The computation time for determining the capacity of one shunting yard was decreased by 17.5%.

Nguyen and Lee (2020) considered the application of DL to TS detection in large Multiple-Input Multiple-Output systems. A DNN architecture for symbol detection is proposed. This network is used in multiple phases of the TS: the initial solution of the LS is approximated by the network, an adaptive early termination algorithm is based on the ANN, and the next candidate search is also based on it. The network is constructed similarly as the DetNet and ScNet architectures and trained by generated samples. The simulation results demonstrate that the proposed algorithm achieves 90% computational complexity reduction for a 32 × 32 system compared to the existing TS algorithms while maintaining almost the same performance.

M. El Alaoui and M. Ettaouil introduce (El Alaoui & Ettaouil, 2021) a CHN to improve the convergence of the SA algorithm for the max-stable problem. The hybrid approach process consists of two phases. The first phase involves a quadratic modelization of CHN for the max-stable problem, similar to an energy function. The second phase involves incorporating CHN into the SA method. The temperature determination process of the SA method takes into account the output of the CHN. It is an important step forward because, in most cases, the ideal cooling rate cannot be determined in advance; it is better to adjust it to the progress of the search. The new approach outperforms other methods in almost all instances when execution time is considered.

Hudson, Li, Malencia, and Prorok (2022) presented a hybrid data-driven approach for solving the Euclidean TSP based on GNN and Guided Local Search (GLS). They define a global regret value of each edge in the problem graph derived from the probability that the given edge is part of an optimal solution or not. A GNN is trained with small-sized problem solutions to approximate this regret value. This model is used by the GLS, which is split into two parts: alternating optimization and perturbation. The LS greedily accepts changes to the solution until a local minimum is reached during the optimization phase. During the perturbation phase, edges in the current solution with high regret are penalized and attempted to be removed by the algorithm, thus enabling it to escape local minima while simultaneously directing it towards promising areas (with low regret) of the solution space. This article gives a new approach to reducing the training time/data requirement. The model has been trained using minor (fast-solving) problems, but in real-world use, it has been used to predict approximate solutions to more significant problems. The experiments show that this approach converges to optimal solutions faster than learning-based and non-learning-based approaches for the TSP. For the 50-node problem set, this leads to 30× times improvement, and for the 100-node problem to 2× improvement. A 7× improvement is achieved when generalizing from 20-node instances to the 100-node problem set.

Correia, Worrall, and Bondesan (2022) presented a novel Neural-SA method to solve several well-known NP-hard problems. The main contribution is the optimization of the neighbor proposal distribution, which defines the space of possible transitions from a given solution during the LS. SA was posed as a Markov decision process, which brought it into reinforcement learning, allowing it to define and train the proposal distribution as a policy. This helps the SA to find the next potential solution within the iteration. The Knapsack, Bin Packing, and TSP were used to show the competitive performance of the method, which outperforms the original SA and can get closer to the global minima in all problems. It is also possible to transfer to problems of different sizes and also to perform well on problems up to 40× larger than the ones used for training. The generalization capabilities are also checked, and using the same architecture on different problem spaces is possible.

Manoochehri and Kolahan (2014) presented an ANN and SA-based integrated algorithm for optimizing the river water pollution index at different water treatment plants. The training database contains real-world measurements of physicochemical parameters and the corresponding water pollution index. Independent multiple linear regression and ANN models are designed and trained to explore the relationship between these. Using the regression model estimation, a SA method is used to find the optimal physicochemical parameters to minimize the pollution index. Finally, the optimum independent variable values of the SA were merged with the nonoptimal independent variable values of the ANN method to set the upper and lower limits for the optimization solution. This hybrid ANN-SA method gained the lowest water pollution index values.

Hu, Li, Deng, Zhao, and Li (2023) presented a novel ML-assisted LS method in the field of Network Enhancement Problems to improve the robustness of a given network by modifying its structure. The standard method to find the best network architecture is based on the k-exchange neighborhood search. The novel idea is to use a neural network pre-trained by synthetic networks to select the best candidates from the local neighborhood. This refinement can significantly speed-up the LS.

4. Evaluation

4.1. Enhanced models

4.1.1. Metaheuristics

In the overall analysis of the articles, attention is first given to the metaheuristics used, as shown in Table 1. One research question was identifying which local search algorithms can be efficiently improved using novel methods.

As can be seen from the table, the selected papers have predominantly aimed to improve the SA algorithm with various ML techniques. This focus is predominantly due to the fact that this method is the closest to the simple hill-climbing-like local search but still eliminates its major drawback of getting stuck in local minima. This attribute is generally the justification for the choice of the method: authors are looking for a technique that preserves the speed of LS but can still handle the problems associated with search spaces with many local minima.

The second most common choice was the TS. It is also a popular method and is already sufficiently complex enough to warrant exploration of potentional improvements. Enhancements typically involve choosing a good starting position. However, other improvements have also been made, such as an ML-assisted early-stopping condition (Nguyen & Lee, 2020) or supporting the suggestion of the next step (Bożejko et al., 2016; Nguyen & Lee, 2020).

In addition to these, a minority of other traditional methods, such as the HC (Persson et al., 2006), GLS (Hudson et al., 2022), Whale Optimization (Tian et al., 2023), VNS (Hakimi-Asiabar et al., 2009), Downhill Simplex (Maslov, 2003), and MCA (Bouhouch et al., 2018) are also utilized.

Several articles do not specify the method beyond mentioning it is a local search. Articles using any novel method not corresponding to any well-known algorithms are also included in this group.

4.1.2. Machine learning

The next subject to be examined is the choice of ML methods used in the novel hybrid solutions, as presented in Table 2. In addition to general ML, the search terms included only keywords related to neural networks, resulting in most methods being based on these.

The dominance of DNN is evident, as they were used in most articles. Conventional feed-forward dense neural networks are very

Table 1
Classification according to the used metaheuristics.

Metaheuristic	# of papers	References
Simulated Annealing	26	Abbasi and Mahlooji (2012), Chen and Yang (2002), Correia et al. (2022), El Alaoui and Ettaouil (2021), Fadaei and Setayeshi (2008), Gao and Tian (2007), Grzechca (2011), Huang et al. (2021), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Liu et al. (2013), Lugon Jr et al. (2009), Manoochehri and Kolahan (2014), Mohebi and Sap (2009), Pan et al. (2020), Rajan et al. (2002), Rao (2006), Rao et al. (2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao et al. (2022), Sheng et al. (2021), Su and Chang (2000), Yadav et al. (2020), Yokoi et al. (2023)
Tabu Search	9	Bożejko et al. (2016), El-Bouri et al. (2005), Hajji et al. (2010), Hsu (2004), Jemai and Mellouli (2008), Nguyen and Lee (2020), Rajan (2009), Rajan and Mohan (2007), Rajan et al. (2003)
Local Search	7	Fister et al. (2016), Hao and Yoshimura (2017), Hu et al. (2023), Shao and Kim (2022), van de Ven et al. (2019), Vitali et al. (2021), Yang (2006)
Hill Climbing	1	Persson et al. (2006)
Guided Local Search	1	Hudson et al. (2022)
Whale Optimization	1	Tian et al. (2023)
Variable Neighborhood Search	1	Hakimi-Asiabar et al. (2009)
Downhill Simplex	1	Maslov (2003)
Min-Conflict Algorithm	1	Bouhouch et al. (2018)

Table 2
Classification according to the machine learning methods.

Machine Learning Method	# of papers	References
Feedforward Neural Network	35	Abbasi and Mahlooji (2012), Chen and Yang (2002), Correia et al. (2022), El-Bouri et al. (2005), Fister et al. (2016), Gao and Tian (2007), Hajji et al. (2010), Hao and Yoshimura (2017), Hsu (2004), Hu et al. (2023), Huang et al. (2021), Jemai and Mellouli (2008), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Liu et al. (2013), Lugon Jr et al. (2009), Manoochehri and Kolahan (2014), Nguyen and Lee (2020), Persson et al. (2006), Rajan (2009), Rajan and Mohan (2007), Rajan et al. (2002, 2003), Rao (2006), Rao et al. (2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao et al. (2022), Sheng et al. (2021), Su and Chang (2000), Vitali et al. (2021), Yadav et al. (2020), Yang (2006), Yokoi et al. (2023)
Self-Organizing Map	4	Grzechca (2011), Hakimi-Asiabar et al. (2009), Maslov (2003), Mohebi and Sap (2009)
Hopfield network	3	Bouhouch et al. (2018), El Alaoui and Ettaouil (2021), Fadaei and Setayeshi (2008)
Support Vector Machine	2	Shao and Kim (2022), Vitali et al. (2021)
Graph Neural Network	2	Hudson et al. (2022), van de Ven et al. (2019)
Convolutional Neural Network	1	Shao and Kim (2022)
K-means	1	Tian et al. (2023)
Other methods	1	Bożejko et al. (2016)

efficient for general pattern recognition and function approximation tasks. The papers demonstrate that this general design is highly efficient unless the task requires the construction of a specific architecture.

Interestingly, the next most commonly used model is not a DNN variant (such as convolutional or recurrent networks) but the SOM. This unsupervised learning method efficiently represents data in lower dimensions, and can be effectively combined with several MHs, like SA (Grzechca, 2011; Mohebi & Sap, 2009), VNS (Hakimi-Asiabar et al., 2009), or Downhill Simplex (Maslov, 2003). It is a versatile tool used to replace the fitness function (Maslov, 2003), assist fitness evaluation (Grzechca, 2011), estimate the correct initial state (Mohebi & Sap, 2009), and modify the principle of LS (Hakimi-Asiabar et al., 2009).

The SVM also appear in the papers, which is an excellent alternative in classification. In both use cases (Shao & Kim, 2022; Vitali et al., 2021), it was used to assist the initial initialization of MHs. Shao and Kim (2022) also use a CNN for feature extraction.

The role of CHNs is also noteworthy, as they were used in three articles. This network model implements a memory using learning procedures similar to those of the biological brain. This attribute determines how it can be combined with heuristics: to help the temperature determination of the SA algorithm (El Alaoui & Ettaouil, 2021) or to find good starting positions for the MCA (Bouhouch et al., 2018) or SA (Fadaei & Setayeshi, 2008) algorithms.

GNNs can be effective in cases when the problem involves describing relationships between different points. Typical examples are various pathfinding tasks, such as the TSP (Hudson et al., 2022). This technique can also determine the maximum capacities of shunting yards (van de Ven et al., 2019).

Among the reviewed papers, only one ML method is not based on neural networks (Tian et al., 2023). In this case, the authors use the K-means algorithm for clustering, with the output determining the starting points of the heuristic launched in the next step.

The last group contains methods that cannot be classified under any of the already-known traditional procedures. Bożejko et al. (2016) designed a novel neural mechanism to prevent the looping of the TS algorithm.

4.2. Hybrid approaches

The previous section has shown which techniques can be used to build a new hybrid model. Even more interesting is how these can be assembled into one, which is shown in Table 3.

A review of the articles shows two commonly used assembly solutions. One is the already mentioned metamodeling, where the ML method replaces a more computationally intensive process. In optimization tasks, resource-intensive tasks often appear in specific steps. This step is often the calculation of the fitness function related to the optimization objective. This function, which determines how good an arbitrary potential solution is, can be complex and time-consuming.

Table 3
Classification according to the hybridization idea.

Hybridization method	# of papers	References
Metamodeling	21	Abbasi and Mahlooji (2012), Chen and Yang (2002), Gao and Tian (2007), Hajji et al. (2010), Hsu (2004), Huang et al. (2021), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Manoochehri and Kolahan (2014), Maslov (2003), Pan et al. (2020), Rao (2006), Rao et al. (2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao et al. (2022), Sheng et al. (2021), Su and Chang (2000), Yadav et al. (2020), Yokoi et al. (2023)
Initialization prediction	17	Bouhouch et al. (2018), El-Bouri et al. (2005), Fadaei and Setayeshi (2008), Jemai and Mellouli (2008), Liu et al. (2013), Lugon Jr et al. (2009), Mohebi and Sap (2009), Nguyen and Lee (2020), Rajan (2009), Rajan and Mohan (2007), Rajan et al. (2002, 2003), Rathinam and Kannan (2014), Shao and Kim (2022), Tian et al. (2023), Vitali et al. (2021), Yang (2006)
Next step estimation	7	Bożejko et al. (2016), Correia et al. (2022), Fister et al. (2016), Hu et al. (2023), Hudson et al. (2022), Nguyen and Lee (2020), van de Ven et al. (2019)
Decision to start	2	Hao and Yoshimura (2017), van de Ven et al. (2019)
Alternating methods	2	Hakimi-Asiabar et al. (2009), Persson et al. (2006)
Fitness calculation	1	Grzechca (2011)
Determine boundaries	1	Rathinam and Kannan (2014)
Cooling process	1	El Alaoui and Ettaouil (2021)
Early termination	1	Nguyen and Lee (2020)

Several papers aim for physical optimizations (e.g., heat transfer problems) where the fitness calculation requires a simulation execution. Since this calculation appears in several steps of MHs, such as selecting the best elements and deciding on the next step, its fast calculation is crucial for the overall runtime. Metamodeling attempts to substitute this computationally expensive process with a much faster approximation function. Since these functions are quite complex, various ML methods can be used in this phase.

Metamodeling is based on the following consecutive steps: (a) building a simulator that can mimic the real-word process (b) generating inputs representative of the total input space (c) running the simulator to calculate the output for the given inputs (d) building and training a neural network using these pairs. In the case of successful training, the ANN will be able to predict the output value for a given input. This network is not expected to be as accurate as a full simulation but should give satisfactory results with significantly reduced computational effort.

Another commonly used general technique is to help the initialization phase of MHs using ML. The drawbacks of MHs are worth observing for better understanding: they tend to use random operations, making them vulnerable to certain problems, such as the initialization of the search. In most cases, there is no information about the part of the search space where the local search should be launched, usually resulting in a random starting point. However, an unfavorable initial state can lead to low-quality results. On the other hand, ML can provide a quick approximate result for a given problem, often lacking the necessary accuracy but acceptable as the starting point of the LS. Therefore, a common hybrid implementation involves training a neural network with possible input–output pairs to give approximate results. Subsequently, a MH is launched from this point to fine-tune the results and find the final solution.

As a variant of the previous approach, two papers (Hao & Yoshimura, 2017; van de Ven et al., 2019) use ML to decide whether to start the LS. In these papers, an ANN estimates how well a given candidate will eventually perform after a LS. If this value is promising, the LS is started, which can further refine the solution. If the estimation indicates that it is not worthwhile to use the MH, the procedure moves to the next potential solution.

In addition to these general methods, some specific methods are particularly interesting from this point of view. El Alaoui and Ettaouil (2021) presented a novel idea to improve the SA algorithm. An essential part of the SA algorithm is the cooling process based on a temperature function, indirectly determining how much the system should allow moving to a higher energy level. This function has multiple widely-used implementations applying simple static mathematical functions depending on search parameters like time and actual fitness level. The novel idea of this paper is to use a CHN to help the cooling process. This hybridization makes it possible to control the convergence towards an optimal solution.

Another inherent problem of MHs is defining a stopping condition, the limit when it is no longer worth running the main iteration. This limit is usually given by specifying some task-dependent fixed conditions, but it is often challenging to specify this in advance. Nguyen and Lee (2020) have published a novel idea in this area. They developed an ANN-based adaptive early termination criterion that can stop the local search instances with a low probability of finding the global optimum.

In the work of Rathinam and Kannan (2014), multiple combinations of ANN and SA are presented to solve the optimization problem. The ANN is used to determine minimum and maximum boundaries for the process parameters of the MH. The paper of Grzechca (2011) is similar to the first group because it uses the ANN for fitness calculation, although it is not considered as metamodeling because the mechanisms of his idea are significantly different. It uses a SOM as a feature selector and classifier.

Two papers (Hakimi-Asiabar et al., 2009; Persson et al., 2006) propose novel solutions to further develop the iteration steps of MHs. Both papers use simple local search (VNS and HC) with the following modification: in each iteration, the algorithm performs the common heuristic step, followed by a step suggested by the ML method. Therefore, these papers are placed in a separate category because the ML and MH approaches do not implement sequentially but alternate continuously.

4.3. Training method

4.3.1. Online or offline training

An essential aspect of ML methods is the training of the selected model, raising several questions. Two factors of the teaching methodology were considered: whether the training was done online or offline (see Table 4), and where the teaching data was obtained from (see Table 5). As seen from the sum of the numbers in the table, some authors do not specify the training method.

In the case of offline training, the application of the ML model is preceded by a completely independent learning phase. The model is trained on a pre-built sample database and later used for predictions to support heuristics. An essential feature of this method is that once the model has been trained, it does not change in the future. This way is considered the most popular and predominantly used in the reviewed articles.

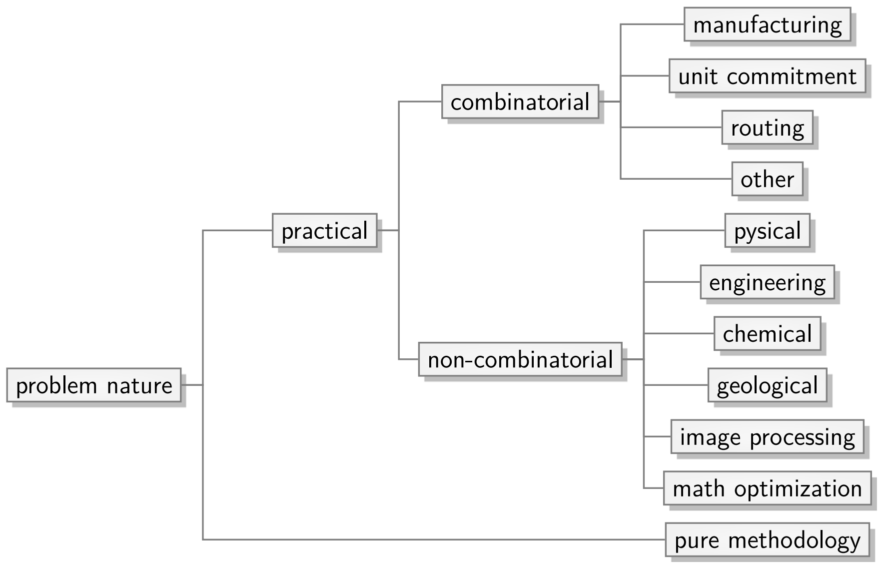

Fig. 3. Topology of problem nature.

Table 4
Classification according to the training methods.

Training method	# of papers	References
Offline	32	Abbasi and Mahlooji (2012), Chen and Yang (2002), Correia et al. (2022), El-Bouri et al. (2005), Hajji et al. (2010), Hsu (2004), Hu et al. (2023), Huang et al. (2021), Hudson et al. (2022), Jemai and Mellouli (2008), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Lugon Jr et al. (2009), Manoochehri and Kolahan (2014), Maslov (2003), Nguyen and Lee (2020), Pan et al. (2020), Rajan et al. (2002, 2003), Rao (2006), Rao et al. (2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao and Kim (2022), Shao et al. (2022), Sheng et al. (2021), Su and Chang (2000), van de Ven et al. (2019), Vitali et al. (2021), Yadav et al. (2020), Yokoi et al. (2023)
Online	6	Bożejko et al. (2016), Fister et al. (2016), Grzechca (2011), Hakimi-Asiabar et al. (2009), Hao and Yoshimura (2017), Persson et al. (2006)

In contrast, implementations where the training process occurs simultaneously with the execution of the heuristic are defined as online training. These online methods require some novel ideas and mostly appear in theoretical works. Fister et al. (2016) use an ANN-based regression to help the LS using online training; meanwhile, the training of the model is embedded into the main iteration of the search. It is also worth highlighting the work of Hao and Yoshimura (2017) who have specifically focused on online learning: they have developed a method to speed-up local search and ran it on practical examples with excellent results.

The chosen neural network architecture significantly influences the way of training also: the traditional feed-forward networks work most straightforwardly with prior supervised learning based on backpropagation, which results in a network with a good estimation ability. In contrast, SOM networks are more similar to a memory and therefore require a different kind of training. Consequently, online learning is used in two SOM-based solutions (Grzechca, 2011; Hakimi-Asiabar et al., 2009).

When choosing the machine learning method, the work of Bożejko et al. (2016) already appeared as an exception, as they developed a completely new neural model which requires online teaching.

The work of Persson et al. (2006) is also unique as they have developed an alternating metaheuristics/ANN solution. They use online learning to increase the accuracy of the neural network continuously.

4.3.2. Source of training data

The subsequent analysis examines the source of the data used to train each ML model, as detailed in Table 5. The most fundamental approach involves using a teaching database entirely drawn from the real world. In some papers, these were already existing databases, or the authors themselves conducted experiments to collect the necessary data for training. In some cases (Hajji et al., 2010; Khosravinia & Kiani, 2023; Sheng et al., 2021), the data collected from real experiments have been augmented with additional generated data.

However, obtaining real-world data presents several challenges. Experiments can be expensive, and the amount of data available may be insufficient. Therefore, training datasets are frequently generated artificially, as explained in the section about metamodeling.

Many articles have dealt with solving various theoretical problems. Since there is no practical background behind such studies that can be measured experimentally, only generated data can be used in these cases. These solutions are similar to the previous one; however, they should be placed in a separate category since they are not based on a simulation that approximates a real-world process but on theoretical functions calculating input–output pairs.

It is also a good idea that the ML model is trained using the solution of small-sized problems but later is used to predict approximate solutions for real-work problem sizes (Correia et al., 2022; El-Bouri et al., 2005; Hudson et al., 2022; Shao & Kim, 2022).

Online training methods should also categorized separately here because training is not done using a pre-generated database but directly at runtime based on the search space.

4.4. Nature of problem

Table 6 classifies the articles into 11 groups based on the specific problem addressed. The classification shown in Fig. 3 is arbitrary, and obviously, many other classifications could have been used. The purpose is merely to give a broad outline of the use cases for the existing methods.

Table 5
Classification according to the source of training data.

Data source	# of papers	References
Real-word experiments	11	Hajji et al. (2010), Hsu (2004), Khosravinia and Kiani (2023), Manoochehri and Kolahan (2014), Pan et al. (2020), Rajan et al. (2002, 2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao et al. (2022), Sheng et al. (2021)
Metamodeling	12	Hajji et al. (2010), Huang et al. (2021), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Lugon Jr et al. (2009), Rao (2006), Rao et al. (2003), Sheng et al. (2021), van de Ven et al. (2019), Yadav et al. (2020), Yokoi et al. (2023)
Theoretical	11	Abbasi and Mahlooji (2012), Chen and Yang (2002), Correia et al. (2022), El-Bouri et al. (2005), Hu et al. (2023), Hudson et al. (2022), Jemai and Mellouli (2008), Nguyen and Lee (2020), Shao and Kim (2022), Su and Chang (2000), Vitali et al. (2021)
On-line	6	Bożejko et al. (2016), Fister et al. (2016), Grzechca (2011), Hakimi-Asiabar et al. (2009), Hao and Yoshimura (2017), Persson et al. (2006)

Table 6
Classification according to the nature of problem.

Problem nature		# of papers	References
combinatorial	manufacturing	6	Bożejko et al. (2016), Chen and Yang (2002), El-Bouri et al. (2005), Shao and Kim (2022), Tian et al. (2023), Yang (2006)
	unit commitment	6	Hajji et al. (2010), Rajan (2009), Rajan and Mohan (2007), Rajan et al. (2002, 2003), Rao et al. (2003)
	routing	4	Gao and Tian (2007), Hudson et al. (2022), Jemai and Mellouli (2008), Vitali et al. (2021)
	other	3	Fadaei and Setayeshi (2008), Hu et al. (2023), van de Ven et al. (2019)
non combinatorial	physical	8	Hsu (2004), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liu et al. (2013), Lugon Jr et al. (2009), Manoochehri and Kolahan (2014), Rathinam and Kannan (2014), Yadav et al. (2020)
	engineering	5	Grzechca (2011), Hao and Yoshimura (2017), Huang et al. (2021), Liao et al. (2023), Nguyen and Lee (2020)
	chemical	3	Pan et al. (2020), Sheng et al. (2021), Yokoi et al. (2023)
	geology	3	Rao (2006), Sen et al. (2021), Shao et al. (2022)
	image processing	1	Maslov (2003)
	math optimization	4	Abbasi and Mahlooji (2012), Bouhouch et al. (2018), El Alaoui and Ettaouil (2021), Mohebi and Sap (2009)
pure methodology		5	Correia et al. (2022), Fister et al. (2016), Hakimi-Asiabar et al. (2009), Persson et al. (2006), Su and Chang (2000)

Some groups relate to solving combinatorial problems, such as scheduling tasks containing several ML-MH-based hybrid solutions. Among these, manufacturing scheduling is also a frequent topic. The unit-commitment problem, though similar, focuses more on efficiently operating a given number of devices. The same can be said of routing problems, which can be either mobile routing algorithms or solutions to the TSP. Three other articles, solving combinatorial problems, are in separate groups: determining the capacity of shunting yard (van de Ven et al., 2019), the optimization of loading patterns (Fadaei & Setayeshi, 2008), and improving the robustness of a given network by modifying its structure (Hu et al., 2023).

Among non-combinatorial tasks, the examined hybrid tools often solve physical problems (e.g. heat transfer). In these cases, metamodeling is typically used to accelerate some complex (often FEM-based) simulation using ML. There are articles from the engineering field (circuit design, antenna design), the chemical field (chemical manufacturing, atomics), and the geological field (water resources). There is one paper from the field of image processing (Maslov, 2003). Another group focuses on optimizations with a solid mathematical background (without practical usage), including one that improves clustering (Mohebi & Sap, 2009).

Lastly, a separate group of articles addresses methodological improvements of MHs. These often use various benchmark functions to validate the improvement.

One research question was whether there are new theoretical advances in the field or if the articles mainly solve practical problems using well-known procedures. Accordingly, the articles are grouped into two categories (as shown in Table 7), although this classification may be imperfect, as in most cases, some combination of theoretical and practical results is presented. In some cases, the decision was obvious (an article is considered theoretical when there is no real-world problem to be solved, and the method is validated by traditional benchmark functions), but in most cases the classification is based on subjective decision.

The classification is based on (a) the generalization ability of the new method presented in the article; (b) the evaluation of the proposed solution: a practical paper mainly evaluates whether the original real-world problem is solved satisfactory or not; a theoretical paper focuses on the comparison with alternative methods taking the given problem only as an example.

4.5. Quantitative analysis

In this research, the quantitative analysis can only be carried out under certain limitations. As seen in Section 4.4, most publications address quite different problems, making straight comparisons impossible. It is only feasible to compare the novel solutions to a given problem with the best existing solutions to the same problem and evaluate the degree of improvement. The evaluation can be based on speed and/or accuracy improvements, depending the data published.

Many articles contain precise numerical data, but the different contexts make this precision less meaningful. Therefore, the articles are grouped into the following categories:

- Speed-up categories: The runtime can be measured in seconds or in iterations; the ratio of the base to the novel method is considered the speed-up.
  - "Minor decrement": the innovative method is usually slower than the traditional solutions (< 100%);

Table 7
Classification according to the nature of novel result.

Nature of result	# of papers	References
Practical	30	Fadaei and Setayeshi (2008), Gao and Tian (2007), Grzechca (2011), Hajji et al. (2010), Hao and Yoshimura (2017), Hsu (2004), Hu et al. (2023), Huang et al. (2021), Khosravinia and Kiani (2023), Lahiri and Chakravorti (2005), Liao et al. (2023), Liu et al. (2013), Lugon Jr et al. (2009), Manoochehri and Kolahan (2014), Maslov (2003), Nguyen and Lee (2020), Pan et al. (2020), Rajan (2009), Rajan and Mohan (2007), Rajan et al. (2002, 2003), Rao (2006), Rao et al. (2003), Rathinam and Kannan (2014), Sen et al. (2021), Shao et al. (2022), Sheng et al. (2021), van de Ven et al. (2019), Yadav et al. (2020), Yokoi et al. (2023)
Theoretical	18	Abbasi and Mahlooji (2012), Bouhouch et al. (2018), Bożejko et al. (2016), Chen and Yang (2002), Correia et al. (2022), El Alaoui and Ettaouil (2021), El-Bouri et al. (2005), Fister et al. (2016), Hakimi-Asiabar et al. (2009), Hudson et al. (2022), Jemai and Mellouli (2008), Mohebi and Sap (2009), Persson et al. (2006), Shao and Kim (2022), Su and Chang (2000), Tian et al. (2023), Vitali et al. (2021), Yang (2006)

 – “Minor increment”: the speed of the innovative method is similar to the traditional solutions, or it is slightly faster (100%–200%);
 – “Major increment”: the novel method is 2-10× faster than the already existing best (200%–1000%);
 – “Significant increment”: the proposed method is more than 10× faster (> 1000%).

- Solution quality categories: this represents the difference in fitness or, at a higher level, the goodness of the solution achieved.
 – “Minor decrement”: the results of the novel method are slightly worse than the base method (< 100%);
 – “Similar”: the accuracy of the method is similar to the state-of-the-art solution (98%–102%);
 – “Minor increment”: the solutions given by the novel method are usually better (100%–200%);
 – “Major increment”: the proposed method is more than 2× accurate (> 200%).

Table 8 includes both classifications, but due to the different problems and solution approaches, it only establishes rough trends. Results are grouped into the main categories given by the hybridization idea: papers about metamodeling (presented in Section 3.2), where the role of ML is to speed-up the fitness calculation by an estimating model; papers in Section 3.1, where ML helps to initialize the local search; and the articles of Section 3.3, which do not fall into any of the previous categories.

All publications are successful because they have either increased speed or accuracy. This encourages the development of hybrid solutions, which can be well used in practice. Authors often focus on increasing either accuracy or speed, ignoring the other parameter. However, many novel solutions are emerging where both characteristics have been improved. The “symmetry” in the table shows the trade-off between the two directions. Achieving extremely high speed usually comes at the expense of reduced accuracy.

This effect can be seen in some metamodeling-based articles, where the fitness computation time is almost zero due to the ML assistance, leading to an outstanding speed-up (Chen & Yang, 2002; Yadav et al., 2020). However, this may come at the cost of a minimal loss in accuracy (Yadav et al., 2020). Authors usually try to find the limit where the quality of the solution does not degrade significantly (perhaps improves slightly), and the corresponding speed-up is maximal.

5. Conclusions

This study explores the field of ML-assisted local searches, presents the results achieved so far, and examines possible research directions. A P.R.I.S.M.A. methodology review was conducted: articles published after 2000 corresponding to the above topic were collected from Scopus and WoS databases. The relevant articles were screened, presented in detail, and evaluated in consecutive steps. Thus, the answer to the first question is affirmative: there have been significant achievements in the research area of ML-assisted local searches.

Looking through the articles, it can be seen that hybrid solutions are mostly based on a combination of the advantages of both approaches:

- Machine learning models can provide very fast approximate answers after a preliminary training.
- Due to their iterative nature, metaheuristics are able to achieve high accuracy and do not require training and the necessary data.

These two features explain how these methods are combined. A common technique is metamodeling, where ML replaces a complex, computationally intensive process (typically the fitness calculation). Another common hybrid variant is when the ML method provides an approximate solution that the local MH tries to refine. These are the two most widely used methods, but other creative solutions have also emerged, such as determining the next step and early stopping of the algorithm.

Regarding the methods used, ANNs are the most commonly employed in ML, particularly deep feed-forward networks. Additionally, some concepts based on SOM, CHN, SVM, and GNN are also presented. From the MH perspective, TS and SA are the most frequently used, as well as other LS solutions in some examples.

The training methods for the ML model were also examined. In most articles, the authors used offline learning as they prepared the neural network on a pre-built training database, which was subsequently used for prediction only. However, innovative solutions that performed the online training during the search were also published in some cases.

In general, a large training dataset is needed, which is provided by real-world measurements in a few articles. A more widely used solution (especially in metamodeling) is to use simulators to artificially generate the data, which are then used to train the models. This is a very efficient solution, especially for inverse problems.

For most articles, it is difficult to determine whether the results are practical or theoretical. However, the evaluation used in this paper indicates that these hybrids are primarily used to solve practical problems, although some entirely novel theoretical novelties are also emerging.

These novel modifications appear even in the most straightforward HC algorithm: there are examples where machine learning is used to determine the next step, but there are also approaches that alternate between traditional heuristic steps and positions recommended by an ANN model. Another innovative development is a model that predicts the quality of possible directions before moving forward, and only explores promising candidates in detail. There are also innovative alternatives for more complex local searches; for example, as an improvement of the TS algorithm, it is possible to use a ML method to replace the traditional tabu list. Interestingly, many of the solutions are based on the use of SOM, which is a less commonly used tool in current mainstream ML research, but works very well here.

Summarizing the results, it is clear that metamodeling and ML-assisted local search initialization have proven to work well and are considered general methods in practice. On the purely theoretical side,

Table 8
Overview of quantitative results (“-”: minor decrement, “0”: similar, “+”: minor increment, “++”: major increment, “+++”: significant increment).

Hybridization idea	Paper	Speed				Solution quality			
		–	+	++	+++	–	0	+	++
metamodeling	Yadav et al. (2020)				∘	∘			
	Chen and Yang (2002)				∘				
	Lahiri and Chakravorti (2005)			∘					
	Gao and Tian (2007)			∘				∘	
	Maslov (2003)		∘				∘		
	Shao et al. (2022)	∘						∘	
	Abbasi and Mahlooji (2012)							∘	
	Huang et al. (2021)							∘	
	Rathinam and Kannan (2014)							∘	
	Hajji et al. (2010)							∘	
	Hsu (2004)								∘
	Khosravinia and Kiani (2023)								∘
initialization	Rao (2006)				∘	∘			
	Yang (2006)			∘				∘	
	Vitali et al. (2021)			∘					∘
	Bouhouch et al. (2018)			∘				∘	
	Rajan (2009)		∘					∘	
	Rajan and Mohan (2007)		∘					∘	
	Lugon Jr et al. (2009)					∘			
	El-Bouri et al. (2005)						∘		
	Shao and Kim (2022)							∘	
	Jemai and Mellouli (2008)								∘
	Fadaei and Setayeshi (2008)								∘
other variants	Manoochehri and Kolahan (2014)						∘		
	Nguyen and Lee (2020)				∘	∘			
	Hao and Yoshimura (2017)			∘		∘			
	Hudson et al. (2022)			∘					
	Fister et al. (2016)		∘					∘	
	van de Ven et al. (2019)		∘						
	Hakimi-Asiabar et al. (2009)		∘					∘	
	Correia et al. (2022)		∘					∘	
	Persson et al. (2006)							∘	
	Bożejko et al. (2016)								∘
	Σ	1	7	7	4	5	3	15	6

however, only a few authors are trying to develop innovative, novel hybrid methods that could be used for general problem-solving.

An exciting and unexplored research direction is modifying the internal functionality of heuristics using ML. There are some attempts (determining the next step, early stopping), but there may be a lot more potential in this area. In addition to specific practical solutions, it would be worthwhile to develop novel solutions that can generally replace existing metaheuristics.

CRediT authorship contribution statement

Sándor Szénási: Conceptualization, Methodology, Data curation, Formal analysis, Writing – original draft, Supervision. **Gábor Légrádi:** Validation, Writing – reviewing & editing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank the “Doctoral School of Applied Informatics and Applied Mathematics” and the “High Performance Computing Research Group” of Óbuda University for their valuable support.

References

Abbasi, B., & Mahlooji, H. (2012). Improving response surface methodology by using artificial neural network and simulated annealing. *Expert Systems with Applications*, *39*(3), 3461–3468.

Bouhouch, A., Bennis, H., Loqman, C., & El Qadi, A. (2018). Neural network and local search to solve binary CSP. *Indonesian Journal of Electrical Engineering and Computer Science*, *10*(3), 1319–1330.

Bożejko, W., Gnatowski, A., Niżyński, T., & Wodecki, M. (2016). Tabu search algorithm with neural tabu mechanism for the cyclic job shop problem. In *Artificial intelligence and soft computing: 15th international conference, ICAISC 2016, zakopane, Poland, June 12–16, 2016, proceedings, part II 15* (pp. 409–418). Springer.

Chen, M.-C., & Yang, T. (2002). Design of manufacturing systems by a hybrid approach with neural network metamodelling and stochastic local search. *International Journal of Production Research*, *40*(1), 71–92.

Correia, A. H. C., Worrall, D. E., & Bondesan, R. (2022). Neural simulated annealing.

da Costa Oliveira, A. L., Britto, A., & Gusmão, R. (2023). Machine learning enhancing metaheuristics: a systematic review. *Soft Computing*, *27*(21), 15971–15998.

El Alaoui, M., & Ettaouil, M. (2021). An adaptive hybrid approach: combining neural networks and simulated annealing to calculate the equilibrium point in max-stable problem. *IAENG International Journal of Computer Science*, *48*(4), 893–898.

El-Bouri, A., Balakrishnan, S., & Popplewell, N. (2005). A neural network to enhance local search in the permutation flowshop. *Computers & Industrial Engineering*, *49*(1), 182–196.

Fadaei, A., & Setayeshi, S. (2008). LONSA as a tool for loading pattern optimization for VVER-1000 using synergy of a neural network and simulated annealing. *Annals of Nuclear Energy*, *35*(10), 1968–1973.

Fister, I., Suganthan, P. N., Fister, I., Kamal, S. M., Al-Marzouki, F. M., Perc, M., et al. (2016). Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution. *Nonlinear Dynamics*, *84*, 895–914.

Gao, M., & Tian, J. (2007). Path planning for mobile robot based on improved simulated annealing artificial neural network. In *Third international conference on natural computation*: *Vol. 3*, (pp. 8–12). IEEE.

Grzechca, D. (2011). Simulated annealing with artificial neural network fitness function for ECG amplifier testing. In *2011 20th European conference on circuit theory and design* (pp. 49–52). IEEE.

Hajji, M., Fares, A., Glover, F., & Driss, O. (2010). Water pump scheduling system using scatter search, tabu search and neural networks—The case of bouregreg water system in Morocco. In *World environmental and water resources congress 2010: challenges of change* (pp. 822–832).

Hakimi-Asiabar, M., Ghodsypour, S. H., & Kerachian, R. (2009). Multi-objective genetic local search algorithm using Kohonen's neural map. *Computers & Industrial Engineering*, *56*(4), 1566–1576.

Hao, C., & Yoshimura, T. (2017). Application of on-line machine learning in optimization algorithms: A case study for local search. In *2017 9th computer science and electronic engineering* (pp. 19–24). IEEE.

Hsu, C.-M. (2004). An integrated approach to enhance the optical performance of couplers based on neural networks, desirability functions and tabu search. *International Journal of Production Economics*, *92*(3), 241–254.

Hu, Q., Li, R., Deng, Q., Zhao, Y., & Li, R. (2023). Enhancing network by reinforcement learning and neural confined local search. In *Proceedings of the thirty-second international joint conference on artificial intelligence* (pp. 2122–2132).

Huang, J., Li, W., He, Y., Zhang, L., & Wong, S.-W. (2021). Optimization of antenna design using the artificial neural network and the simulated annealing algorithm. In *2021 computing, communications and ioT applications* (pp. 119–122). IEEE.

Hudson, B., Li, Q., Malencia, M., & Prorok, A. (2022). Graph neural network guided local search for the traveling salesperson problem. In *International conference on learning representations*. [Online]. Available: https://openreview.net/forum?id=ar92oEosBIg.

Jemai, J., & Mellouli, K. (2008). A neural-tabu search heuristic for the real time vehicle routing problem. *Journal of Mathematical Modelling and Algorithms*, *7*, 161–176.

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E.-G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, *296*(2), 393–422.

Khosravinia, K., & Kiani, A. (2023). Optimizing the ultrashort laser pulses for in situ nanostructure generation technique for high-performance supercapacitor electrodes using artificial neural networks and simulated annealing algorithms. *ACS omega*, *8*(19), 17220–17233.

Lahiri, A., & Chakravorti, S. (2005). A novel approach based on simulated annealing coupled to artificial neural network for 3-D electric-field optimization. *IEEE Transactions on Power Delivery*, *20*(3), 2144–2152.

Liao, Y., Yu, T., Wang, Y., Dong, B., & Yang, G. (2023). Simulated annealing algorithm with neural network for designing topological photonic crystals. *Optics Express*, *31*(19), 31597–31609.

Liu, W., Ye, Y.-L., Si, F., & Xu, Z.-G. (2013). Determination of the optimum initial operation pressure of a steam turbine unit based on a BP(back propagation) neural network and SA-BBO(simulated annealing biogeography-based optimization) algorithm. *Journal of Engineering for Thermal Energy and Power*, *28*, 18–22.

Lugon Jr, J., Silva Neto, A. J., & Santana, C. C. (2009). A hybrid approach with artificial neural networks, Levenberg–Marquardt and simulated annealing methods for the solution of gas–liquid adsorption inverse problems. *Inverse Problems in Science and Engineering*, *17*(1), 85–96.

Manoochehri, M., & Kolahan, F. (2014). Integration of artificial neural network and simulated annealing algorithm to optimize deep drawing process. *International Journal of Advanced Manufacturing Technology*, *73*, 241–249.

Maslov, I. V. (2003). Improving local search with neural network in image registration with the hybrid evolutionary algorithm. In *Intelligent computing: theory and applications*: *Vol. 5103*, (pp. 166–177). SPIE.

Mohebi, E., & Sap, M. (2009). An optimized hybrid kohonen neural network for ambiguity detection in cluster analysis using simulated annealing. In *Enterprise information systems: 11th international conference, ICEIS 2009, milan, Italy, May 6–10, 2009. proceedings 11* (pp. 389–401). Springer.

Nguyen, N. T., & Lee, K. (2020). Deep learning-aided tabu search detection for large MIMO systems. *IEEE Transactions on Wireless Communication*, *19*(6), 4262–4275.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., et al. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *International Journal of Surgery*, *88*, Article 105906.

Pan, R., Duque, J. V. F., Martins, M. F., & Debenest, G. (2020). Application of a neural fuzzy model combined with simulated annealing algorithm to predict optimal conditions for polyethylene waste non-isothermal pyrolysis. *Heliyon*, *6*(11).

Persson, A., Grimm, H., & Ng, A. H. (2006). Simulation-based optimisation using local search and neural network metamodels. In *Artificial intelligence and soft computing* (pp. 178–183). Citeseer.

Rajan, A. C. G. C. (2009). Neural based tabu search method for solving unit commitment problem with cooling-banking constraints. *Serbian Journal of Electrical Engineering*, *6*(1), 57–74.

Rajan, C. C. A., & Mohan, M. (2007). Neural-based tabu search method for solving unit commitment problem for utility system. *International Journal of Energy Technology and Policy*, *5*(4), 489–508.

Rajan, C. A., Mohan, M., & Manivannan, K. (2002). Neural-based simulated annealing method for solving the unit commitment problem. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, *10*(4), 241–254.

Rajan, C. A., Mohan, M., & Manivannan, K. (2003). Neural-based tabu search method for solving unit commitment problem. *IEE Proceedings-Generation, Transmission and Distribution*, *150*(4), 469–474.

Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review*, 1–71.

Rao, S. (2006). A computationally efficient technique for source identification problems in three-dimensional aquifer systems using neural networks and simulated annealing. *Environmental Forensics*, *7*(3), 233–240.

Rao, S., Thandaveswara, B., Murty Bhallamudi, S., & Srinivasulu, V. (2003). Optimal groundwater management in deltaic regions using simulated annealing and neural networks. *Water Resources Management*, *17*, 409–428.

Rathinam, V., & Kannan, T. (2014). Optimization of ferrite number of solution annealed duplex stainless steel CladdingUsing integrated artificial neural network: Simulated annealing. *Research Journal of Applied Sciences, Engineering and Technology*, *7*(21), 4464–4475.

Sen, S., Rai, A., Chakrabarty, J., Lahiri, S. K., & Dutta, S. (2021). Parametric modeling and optimization of phycoremediation of Cr (VI) using artificial neural network and simulated annealing. *Algae: Multifarious Applications for a Sustainable World*, 103–133.

Shao, X., & Kim, C. S. (2022). An adaptive job shop scheduler using multilevel convolutional neural network and iterative local search. *IEEE Access*, *10*, 88079–88092.

Shao, S., Su, B., Zhang, Y., Gao, C., Zhang, M., Zhang, H., et al. (2022). Sample design optimization for soil mapping using improved artificial neural networks and simulated annealing. *Geoderma*, *413*, Article 115749.

Sheng, G., Sun, M., & Liu, H. (2021). Exploring the effect of reaction conditions on the yield of C4 olefins based on neural networks and improved simulated annealing. In *2021 IEEE 3rd international conference on frontiers technology of information and computer* (pp. 86–90). IEEE.

Su, C.-T., & Chang, H.-H. (2000). Optimization of parameter design: an intelligent approach using neural network and simulated annealing. *International Journal of Systems Science*, *31*(12), 1543–1549.

Talbi, E.-G. (2021). Machine learning into metaheuristics: A survey and taxonomy. *ACM Computing Surveys*, *54*(6), 1–32.

Tian, G., Zhang, C., Zhang, X., Feng, Y., Yuan, G., Peng, T., et al. (2023). Multi-objective evolutionary algorithm with machine learning and local search for an energy-efficient disassembly line balancing problem in remanufacturing. *Journal of Manufacturing Science and Engineering*, *145*(5), Article 051002.

van de Ven, A., Zhang, Y., & Lee, W.-J. (2019). Boosting local search using machine learning: a study on improving local search by graph classification in determining capacity of shunting yards. In *Agents and artificial intelligence: 11th international conference, ICAART 2019, prague, czech Republic, February 19–21, 2019, revised selected papers 11* (pp. 183–203). Springer.

Vitali, T., Mele, U. J., Gambardella, L. M., & Montemanni, R. (2021). Machine learning constructives and local searches for the travelling salesman problem. In *International conference on operations research* (pp. 59–65). Springer.

Yadav, R., Tripathi, S., Asati, S., & Das, M. K. (2020). A combined neural network and simulated annealing based inverse technique to optimize the heat source control parameters in heat treatment furnaces. *Inverse Problems in Science and Engineering*, *28*(9), 1265–1286.

Yang, S. (2006). Job-shop scheduling with an adaptive neural network and local search hybrid approach. In *The 2006 IEEE international joint conference on neural network proceedings* (pp. 2720–2727). IEEE.

Yokoi, T., Kato, H., Oshima, Y., & Matsunaga, K. (2023). Atomic structures of grain boundaries for Si and Ge: A simulated annealing method with artificial-neural-network interatomic potentials. *Journal of Physics and Chemistry of Solids*, *173*, Article 111114.