



## SemiACO: A semi-supervised feature selection based on ant colony optimization

Fereshteh Karimi, Mohammad Bagher Dowlatshahi\*, Amin Hashemi

*Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khoramabad, Iran*



### ARTICLE INFO

**Keywords:**

Semi-supervised Feature Selection  
Ant Colony Optimization  
Reinforcement Learning  
Nonlinear Heuristic Function  
Temporal difference

### ABSTRACT

Feature selection is one of the most efficient procedures for reducing the dimensionality of high-dimensional data by choosing a practical subset of features. Since labeled samples are not always available and labeling data may be time-consuming or costly, the importance of semi-supervised learning becomes apparent. Semi-supervised learning deals with data that includes both labeled and unlabeled instances. This article proposes a method based on Ant Colony Optimization (ACO) for the semi-supervised feature selection problem called SemiACO. The SemiACO algorithm finds features by considering the minimum redundancy between features and the maximum relevancy between the features and the class label. The SemiACO uses a nonlinear heuristic function instead of a linear one. The heuristic learning technique for the ACO heuristic function utilize a Temporal Difference (TD) reinforcement learning algorithm. We characterize the feature selection search space as a Markov Decision Process (MDP), where features indicate the states, and selecting the unvisited features by each ant represents a set of actions. We contrast the efficiency of SemiACO based on various experiments on 14 benchmark datasets, comparing eight semi-supervised feature selection methods.

### 1. Introduction

Nowadays, data are produced in large amounts and dimensions that can cause problems for data mining and machine learning algorithms, known as the curse of dimensions. The Dimension Curse degrades the accuracy of learning models, causes algorithms to over-fit, and increases computing complexity and learning time. Obtaining useful information from this large amount of data is a critical challenge, and many researchers are trying to deal with this issue by proposing methods to decrease the data dimensionality (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2020; Hashemi, Pajoohan, et al., 2022). Feature selection is one of the most effective dimensionality reduction methods, which selects a subset of the original feature set by eliminating unrelated and redundant features from data (Deng, Li, Weng, & Zhang, 2019; Miri, Dowlatshahi, & Hashemi, 2022a; Solorio-Fernández, Carrasco-Ochoa, & Martínez-Trinidad, 2020).

Based on interaction with the learning algorithms, feature selection methods are classified into three categories: filter, wrapper, and embedded (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2021e; Miri,

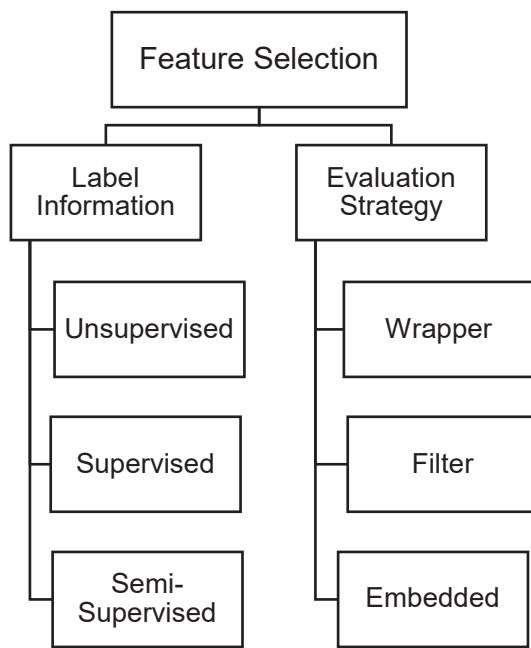
Dowlatshahi, & Hashemi, 2022b). Filter strategies are not dependent on learning algorithms and evaluate the features based on statistical measures (Hashemi & Dowlatshahi, 2022; Miri et al., 2022; Hashemi, Joodaki et al., 2022). A classification algorithm is utilized to assess the possible feature subsets in the wrapper techniques. Embedded techniques perform a learning model once and then evaluate the features similar to filter methods (Hashemi, Dowlatshahi, & Nezamabadi-Pour, 2021c; Miao & Niu, 2016; Venkatesh & Anuradha, 2019; Hashemi, Dowlatshahi, & Nezamabadi-pour, 2022).

Feature selection techniques in another category are classified into three groups based on the availability of the class label: supervised, unsupervised, and semi-supervised (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2021a; Joodaki, Dowlatshahi, & Joodaki, 2021). In supervised feature selection methodologies, the features are assessed according to their level of connection with the class label (Beiranvand, Mehrdad, & Dowlatshahi, 2022; Dalvand, Dowlatshahi, & Hashemi, 2022; Hashemi, Dowlatshahi, & Nezamabadi-pour, 2021e). We have no information about the class label in unsupervised feature selection techniques. In semi-supervised methods, only a tiny percentage of the

Peer review under responsibility of Submissions with the production note 'Please add the Reproducibility Badge for this item' the Badge and the following footnote to be added: The code (and data) in this article has been certified as Reproducible by the CodeOcean: <https://codeocean.com>. More information on the Reproducibility Badge Initiative is available at <https://www.elsevier.com/physicalsciencesandengineering/computerscience/journals..>

\* Corresponding author.

E-mail addresses: [karimi.fr@fe.lu.ac.ir](mailto:karimi.fr@fe.lu.ac.ir) (F. Karimi), [dowlatshahi.mb@lu.ac.ir](mailto:dowlatshahi.mb@lu.ac.ir) (M.B. Dowlatshahi), [hashemi.am@fe.lu.ac.ir](mailto:hashemi.am@fe.lu.ac.ir) (A. Hashemi).



**Fig. 1.** The structure of feature selection methods categorization.

data is labeled (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2021d; Sheikhpour, Sarram, Gharaghani, & Chahooki, 2017; van Engelen & Hoos, 2020). Fig. 1 shows the outline of feature selection methods categorization.

Hybrid optimization problems are NP-hard because they cannot be solved in a limited time by polynomial computation. Therefore, using approximate techniques to solve complex problems is necessary. Many optimization problems can be solved using metaheuristic techniques (Bayati, Dowlatshahi, & Paniri, 2020; Dowlatshahi et al., 2014; 2021; Dowlatshahi & Nezamabadi-pour, 2014). These techniques can quickly achieve near-optimal solutions in a reasonable time (Hashemi, Dowlatshahi, & Nezamabadi-Pour, 2021b; Zhao, He, & Wang, 2021; Zhou et al., 2021).

ACO is a population-based metaheuristic approach that focuses on the behavior of a colony of ants in nature (Hashemi, Joodaki et al., 2022). The procedure of the ACO algorithm is based on a random search, and the decision-making process is usually conducted using a component, a chemical called pheromone, that shows the quality of a solution in the search space (Jia, Kwong, Hou, & Wu, 2020). In ACO, the ants work together to find the optimal path. Each ant randomly travels through existing paths, deposits several pheromones along the path, and updates the pheromone in its path. The optimal path with more pheromone trails will be selected (Ma, Yang, Nie, Uijlings, & Sebe, 2011; Sugiyama, Idé, Nakajima, & Sese, 2010).

Reinforcement learning is a problem-solving technique in machine learning that consists of an agent that learns through trial-and-error interactions with a dynamic environment (Zhao, Zhang, et al., 2021). At every stage of the interaction, the agent chooses an action that modifies the state of the environment. It is then rewarded or penalized for evaluating the quality of transition in each stage based on its performance. Meanwhile, TD is one of the problem-solving approaches in reinforcement learning. The value of a state or action is computed using the value of other states or actions (Reinforcement Learning: A Survey, 1996; Mazyavkina, Sviridov, Ivanov, & Burnaev, 2021).

This article suggests a novel semi-supervised feature selection method that utilizes the ACO algorithm for selecting practical features. As far as we know, no semi-supervised feature selection technique has used ACO. Given that the nonlinear heuristic procedure performs better than the linear one, the proposed algorithm, called SemiACO, uses nonlinear functions instead of linear ones to learn the ACO heuristic

function using reinforcement learning. A nonlinear heuristic function can be utilized to identify significant relationships between features. The reason for using the ACO algorithm for the semi-supervised feature selection problem is that this approach has not been used in this problem until now. Also, the ACO algorithm is a powerful metaheuristic method that has shown significant performance in feature selection problems. The main difference between the proposed algorithm and the former methods in the literature is that the features are not evaluated based on a fixed function. The proposed method uses a heuristic learning strategy, leading to better solutions by learning from the experiences gathered. Heuristic learning aims to improve the heuristic function by learning from experiences and helping an agent to estimate future actions more accurately.

For heuristic learning, we require the integration of the ACO algorithm and reinforcement learning. Therefore, we model the feature selection search space into an MDP, where features indicate the states ( $S$ ). Choosing the unvisited feature by every ant demonstrates a collection of actions. Reward signals are a combination of two criteria when ants take action: The first function is the cosine similarity between the current feature (state) and the other ones (for both labeled and unlabeled samples) to measure the redundancy. The second function is the Maximal Information Coefficient (MIC) between the features and the class label of all labeled instances to assess the relevancy of features.

Furthermore, the pheromone values are updated by the “global updating rule” in the ACO. The state-value function ( $V$ ) is straightly updated by the temporal difference formula to form a learned heuristic function. The proposed algorithm initializes the pheromones value and sequence function by MIC and mutual information among features and the class label considering labeled samples. Eventually, the importance of the features will be specified based on the pheromones value, and by sorting those values in descending order, the desired number of top features will be chosen. The proposed algorithm is a filter-based technique and does not depend on any classifier in the feature selection process.

The main characteristics of the proposed algorithm are as follows:

- This is the first time a semi-supervised feature selection method has been proposed based on ACO.
- Since semi-supervised learning applies to a combination of labeled and unlabeled data samples, the proposed algorithm computes the relevancy between features and class labels of labeled samples. In addition, it calculates the redundancy between the features (all labeled and unlabeled instances). The proposed method uses the information of labeled and unlabeled samples simultaneously.
- The proposed algorithm initializes the value function with a nonlinear heuristic function instead of a linear one.
- The mean execution time is sensible and competitive with the usual semi-supervised feature selection techniques.

To demonstrate the efficiency and optimality of the proposed algorithm, we have compared our method with eight semi-supervised feature selection techniques based on fourteen real-world datasets. The results demonstrate the superiority of the proposed algorithm over the other competitive methods based on well-known classification measures.

The article structure is as follows: Section 2 reviews the related techniques in the literature and introduces the algorithm’s fundamental concepts. Section 3 discusses the proposed algorithm in detail. Section 4 and Section 5 explain the configuration of the experiments and the obtained results, respectively. Eventually, Section 6 presents the conclusions.

## 2. Fundamental concepts and literature review

### 2.1. semi-supervised learning

In semi-supervised learning, a small number of the samples are labeled. In other words, the dataset has two kinds of samples: labeled and unlabeled. Let us assume a dataset containing  $n$  instances,  $d$  features, and  $c$  classes ( $Y$ ), which is divided into two subsets: A set of  $l$  labeled samples as  $X_L = (x_1, \dots, x_l)$  which is assigned to the class label  $Y_L = (y_1, \dots, y_l)$  and a set of  $u = n - l$  unlabelled samples  $X_U = (x_{l+1}, \dots, x_{l+u})$ . All datasets are denoted by  $X = X_L \cup X_U$  and  $Y = Y_L$  (Sheikhpour et al., 2017; van Engelen & Hoos, 2020).

### 2.2. Semi-supervised feature selection

Data produced in many real-world applications have vast numbers of unlabeled data samples and a few labeled ones. In fact, it takes time and effort for the user to label every data sample. In this context, the subset of the labeled samples is usually too small to carry enough information for supervised selection, but unsupervised approaches ignore this labeled information that can still be important for feature selection. Thus, there has been more interest in a new feature selection challenge called the semi-supervised feature selection problem. By considering both labeled and unlabeled data, semi-supervised feature selection methods yield a solution in finding the best subset of features (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2020; Hashemi, Dowlatshahi, & Nezamabadi-Pour, 2021c).

### 2.3. Related works

This part briefly explains some semi-supervised feature selection techniques and related techniques in the feature selection task.

Supervised feature selection techniques serve poorly due to overfitting when just a tiny number of labeled samples are available. In particular cases, unlabeled samples can enhance learning efficiency. Sugiyama et al. (2010) recommended a semi-supervised feature selection method that maintains the global construction of unlabeled instances to separate labeled samples in various categories from each other. SELF stands for Semi-Supervised Fisher's local discriminant analysis, contains an analytical structure of the globally optimal solution, and is calculated based on a specific analysis.

SGFS (Dalvand et al., 2022) is a simple semi-supervised algorithm that uses the concept of PageRank centrality for feature selection. A complete graph is constructed first by the relevancy of features with the class label in labeled data and the redundancy of features considering all labeled data. The features are the nodes in this graph. The PageRank algorithm evaluates this graph to order the features.

The explosive growth of digital images needs effective techniques to handle these images. Automatic image annotation is an effective procedure for image control assignments. Ma et al. (2011) proposed a technique to jointly choose the most relevant features from all the data points by utilizing a sparsity-based model and exploiting labeled and unlabeled data to learn manifold construction. This framework can simultaneously learn a robust classifier for image annotation by choosing the discriminative features related to the semantic conceptions.

Constrained Laplacian Score (CLS) is a method proposed for semi-supervised feature selection and has reported better ratio performance than other methods. Because CLS uses labeled and unlabeled samples' information to choose the most relevant features. Obtaining information from the labeled samples (expressed by paired limitations) is still a critical problem. Constraints are proven to have some noise that may worsen learning efficiency. Benabdeslem and Hindawi (2011) introduced a method based on the ensemble of data resampling (bagging) and a random subspace approach. The technique generates a global ranking

of features by aggregating numerous constraint Laplacian scores based on various views of the available labeled and unlabeled data.

Sheikhpour, Sarram, and Sheikhpour (2018) recommended an iterative structure based on graph Laplacian and mixed convex and non-convex  $l_{2,p}$ -norm ( $0 < p \leq 1$ ) regularization proposed for regression problems. In the proposed framework, a semi-supervised graph Laplacian-founded scatter matrix created for regression problems is utilized for encoding the class label of labeled data and the local construction of both labeled and unlabeled instances.

Jia et al. (2020) used a non-negative matrix factorization (NMF) structure by modeling the label information to find the best subset of features. The proposed method is skilled in making low-dimensional displays to enhance clustering performance. In particular, to guide factorization, a pair of complementary regulators and similarities are included in the conventional NMF. In addition, it limits the similarity and dissimilarity of the low-dimensional demonstrations of labeled instances and tiny unlabeled samples.

All unlabeled data can be assumed as positive or negative points if we have some binary labeled and unlabeled samples. In this approach, academic and practical studies provide strong outcomes for feature selection through hypothesis testing and feature ranking. Sechidis et al. (2018) showed that by integrating them with some earlier "soft" knowledge of the domain, gain two unique techniques (Semi-JMI and Semi-IAMB) that perform remarkably better than the competitive techniques. They perform well when the class label is unavailable, but not by accident.

Liu, Nie, Wu, and Chen (2010) proposed a strategy in which the technique analyzes the distribution of labeled and unlabeled data with a specific label publishing technique for transmitting soft labels of unlabeled data. Then an efficient algorithm for optimizing the tracking ratio measure chooses the optimal feature subset directly. With this technique, the dispensation of labeled and unlabeled samples is investigated by the label propagation method to earn soft labels on unlabeled instances. Then, the trace ratio criterion optimization utilizes directly to select the optimal feature subset.

Evolutionary algorithms have shown promising performances in feature selection tasks in recent years. Song, Zhang, Guo, Sun, and Wang (2020) used coevolutionary particle swarm optimization (PSO) to select the best features in high-dimensional data. This paper employs the "divide and conquer" idea in a cooperative coevolutionary procedure. Chen, Xue, Zhang, and Zhou (2022) utilized an evolutionary multitasking process using a PSO-based algorithm to avoid falling into local optima and high-computational costs. Evolutionary multitasking can help the PSO algorithm to converge faster and increase the global search capability. Also, other evolutionary bases are proposed in the literature using exciting ideas, such as the combination of the PSO algorithm and clustering (Song, Zhang, Gong, & Gao, 2021) and the combination of multiobjective optimization and the concept of steering matrix (Cheng, Chu, Xu, & Zhang, 2021). Ant-MCDM (Hashemi, Joodaki et al., 2022) is another feature selection method that uses an ensemble of heuristics to construct the ACO heuristic values. A multi-criteria decision-making approach is utilized to perform the ensemble process. MSSL (Bayati, Dowlatshahi, & Hashemi, 2022) is a Memetic-based feature selection algorithm for multi-label data. In this method, the gradient descent algorithm performs the local search.

### 2.4. Reinforcement learning

Reinforcement learning is an approach for handling hybrid optimization problems in machine learning, a structure in which the agent learns how to perform successive decision-making tasks online through interaction with the environment. Reinforcement learning in agent planning is provided by receiving feedback on the outcome of choices made as a reward or punishment without specifying how to achieve the outcome. In reinforcement learning, the agent first chooses an action from the limited and possible action collection based on observing a

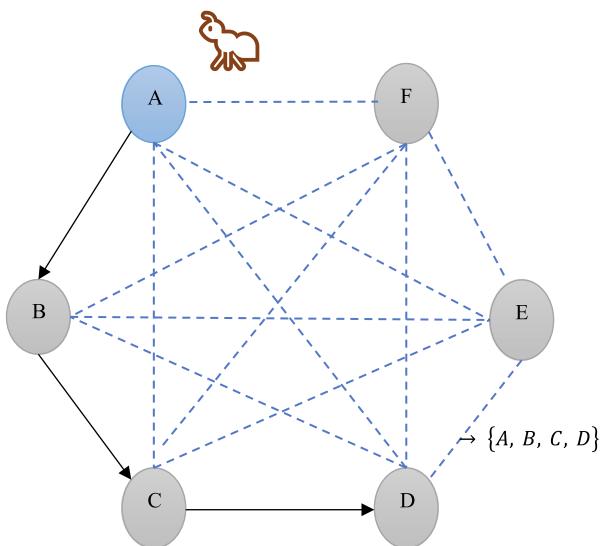


Fig. 2. The display of ACO for feature selection.

situation in the environment and performing that action. Then, the agent receives a predetermined signal from the environment, demonstrating the quality of the operator's action as a reward or punishment. In the next step, the agent transfers to a new environmental status based on the current state (Joshi, Kale, Gandewar, Korate, Patwari, & Patil, 2021; Kober, Bagnell, & Peters, 2013; Mazyavkina et al., 2021). In this approach, the agent interacts with the environment by performing a series of actions to find solutions (Baird, 2020; Sutton, Precup, & Singh, 1999). MDP provides a widely utilized mathematical framework for modeling such problems and consists of a tuple  $\langle S, A, T, R \rangle$  (Baird, 2020; Bäuerle & Rieder, 2010; Sutton et al., 1999):

A set of states,  $S = \{s_1, s_2, \dots, s_d\}$

A set of actions,  $A = \{a_1, a_2, \dots, a_m\}$

A state transfer function  $T = (s'|s, a)$  is a possibility distribution function that a given state  $s$  and action into a state  $s'$ .

A reward function  $R = S \times A \times S \rightarrow \mathbb{R}$  gives an instant reward when an agent performs an action and moves from state to state  $s'$ .

Using the Markov chain in reinforcement learning, the agent's choice of action is subject to a policy that determines the probability of choosing the action in a specific status. In other words, it determines the effect of the action in an independent state in such a way that the reinforcing learning agent learns to maximize all future rewards (Reinforcement Learning: A Survey, 1996; Y. Wang, Wang, Liao, & Chen, 2017).

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1}, \quad (1)$$

where  $t$  is the time stage and  $r_{t+1}, r_{t+2}, r_{t+3}, \dots$  is the sequence of rewards after the time stage  $t$ , and  $\gamma \in [0, 1]$  is a deterrent that handles the significance of instant rewards compared to coming rewards and prevents the reward from going to infinity.

One of the best techniques for solving the Markov decision chain problem is the TD technique, which is notable for its good performance, low computational cost, and plain interpretation. The value of a state or action is estimated using the value of other states or actions (Paniri, Dowlatshahi, & Nezamabadi-pour, 2021; van Seijen, Mahmood, Pilarski, Machado, & Sutton, 2016). Since the proposed technique basis on TD, we express TD as follows:

$$V(S_t) = V(S_t) + \alpha[r_{t+1} + \gamma V(S_{t+1}) - V(S_t)], \quad (2)$$

where parameter  $\alpha$  is the learning rate that determines how many errors

can be accepted at every step. Parameter  $\gamma$  is the discount rate that characterizes the influence of the following case. The value inside the bracket is a calculation error in the calculation. It calculates the difference between the worth of case  $V(S_t)$  and the estimate of the subsequent step and the subsequent reward  $r_{t+1} + \gamma V(S_{t+1}) - V(S_t)$  that the operator tries to minimize this time.

## 2.5. Ant colony optimization

ACO algorithm was introduced by Dorigo et al. (Dorigo & Gambarella, 1997). The ACO is a swarm intelligence metaheuristic and random search technique based on the natural behavior of real ants and is applicable in solving optimization problems. The principal part of the ACO algorithm is the concept of pheromone, which uses the search space for possible sampling. When the ant moves from the nest to the food, it emits a trail of a pheromone chemical. The amount of pheromone evaporates over time, which prevents ants from being trapped in the local optimum. Therefore, the more ants travel on a path, the more intense the pheromone amount and the greater the probability of selecting that as the optimal solution. In addition to the characteristics of real ants, the ACO algorithm adds other capabilities to artificial ants. They have an inner state that holds the record of their previous actions and assesses the goodness of the solution produced by every ant. They update the pheromone sequence locally and globally to gain superior solutions. In addition to pheromone values, it can utilize some heuristic information to conduct the search procedure (Dorigo & Blum, 2005; Monteiro, Fontes, & Fontes, 2012; Paniri et al., 2020, 2021).

The general procedure of the algorithm is that after initializing the parameters: In each iteration, the ants will probably formulate solutions to the problem using a specific pheromone formula until the stop criteria satisfy. When the ants have completed their answer, the pheromone evaporates uniformly in all cases. Every ant then updates the pheromone vector based on its solutions. Algorithm 1 shows the general framework of an ACO algorithm (Dorigo & Blum, 2005; Monteiro et al., 2012). The problem must be formulated appropriately to illustrate the ACO-based feature selection issue. The nodes represent the features, and the edges between nodes are the connection between them. The search for the optimal feature subset is a check-in in which the ant visits the least number of nodes in the graph until the criterion is satisfied. For example, the stop criterion may achieve high accuracy with fewer features than the original features set (Kanan, Faez, & Taheri, 2007).

Algorithm 1. General pseudo-code of the ant colony optimization.

- 
1. Initializing the pheromone trails
  2. Repeat
  3. For every ant, Do
  4. Solution building by modeling the issue in terms of pheromone vector;
  5. Update the pheromone with the established solutions:
  6. Evaporation
  7. Reinforcement
  8. Until stopping criteria
  9. Output: final pheromone vector that characterizes the best-founded solution

Fig. 2 shows this scroll. An ant is currently in node A, and the dotted lines indicate which features can add to its path. According to the transmission rule, ant first chooses feature B, then C, and then when the criterion for stopping is satisfied, the subset  $\{A, B, C, D\}$  is determined. The ant ends its traversal, and this subset provides the feature as a candidate for feature decrease as output.

The ACO creates probabilistic decisions founded on local heuristic information (solution suitability) and the pheromone sequence (record of solutions created). The ants update the pheromones associated with each feature (graph nodes) over time (Paniri, Dowlatshahi, & Nezamabadi-pour, 2020).

In a probabilistic procedure, ant  $k$  utilizes the probabilistic action selection law to transfer from feature  $i$  to feature  $j$  according to the following equation (Paniri et al., 2020):

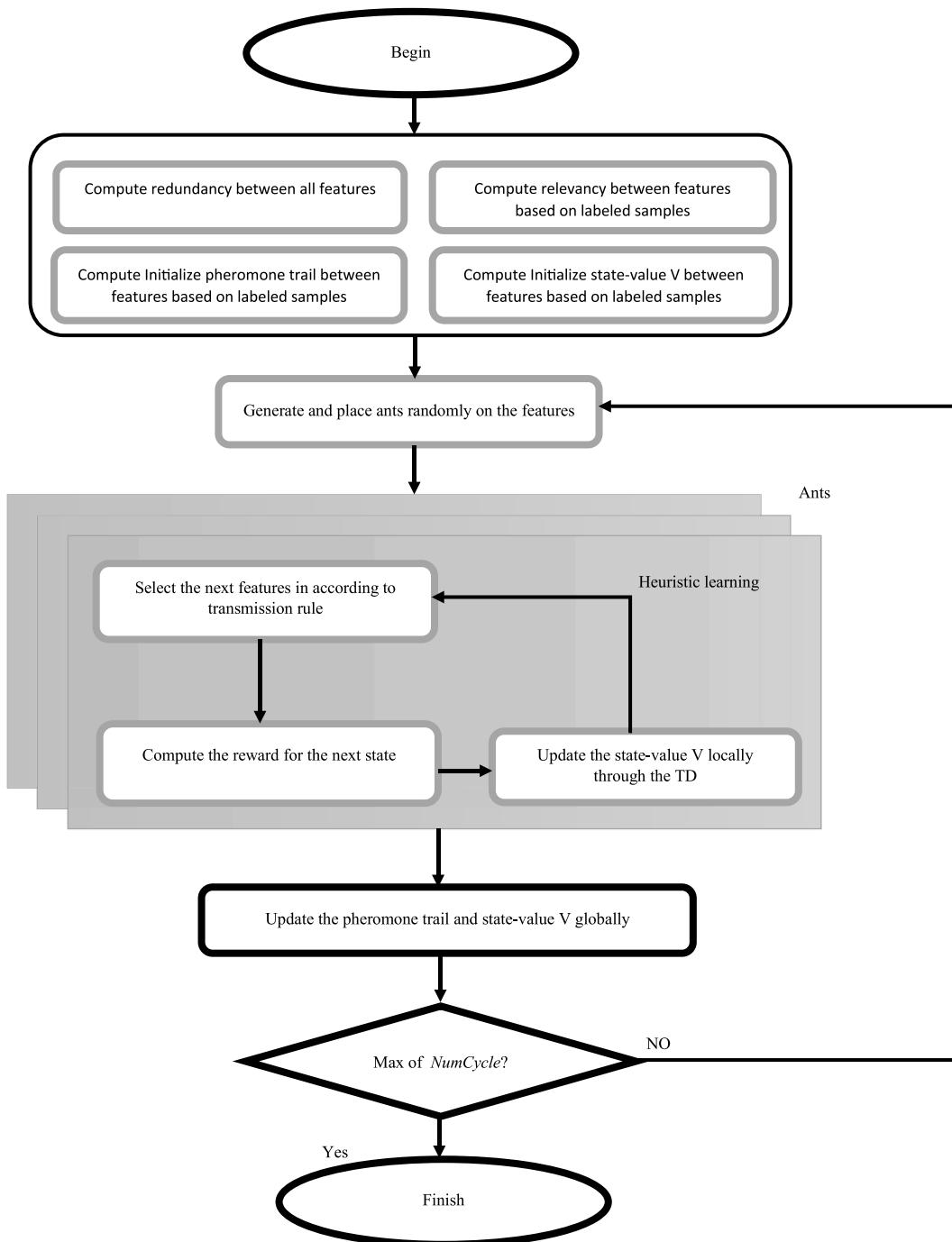


Fig. 3. The general flowchart SemiACO.

$$p_{ij}^k(t) = \begin{cases} [\tau_i(t)] [\eta(F_i, F_j)]^\beta / \sum_{u \in N_i} [\tau_u] [\eta(F_i, F_u)]^\beta, & \forall j \in N_i^k, \text{ if } q > q_0, \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where  $p_{ij}^k(t)$  is a transmission possibility of  $k$ -th ant to move from feature  $i$  to feature  $j$  at time  $t$ ,  $\tau_i$  is the amount of pheromone allocated to feature  $i$ ,  $\eta(F_i, F_j)$  is the heuristic information between feature  $(F_i, F_j)$ .  $N_i^k$  is a collection of neighbor features  $i$ ,  $q$  is an accidental variable that is uniformly distributed in ranges  $[0, 1]$ , and  $q_0$  is a constant variable ( $0 \leq q_0 \leq 1$ ). The value of the parameter  $\beta$  is in the ranges  $[0, 1]$  determines the relative significance of heuristic information and the pheromone value. If  $\beta = 0$ , the heuristic information is ignored, and the

decision is based solely on the previous action history.

Greedy, every ant visits the next feature according to the following equation (Paniri et al., 2020):

$$i = \arg \max_{u \in N^k} \{[\tau_u] [\eta(F_i, F_u)]^\beta\}, \text{ if } q \leq q_0. \quad (4)$$

The parameters  $q_0$  balance the exchange between the exploitation and exploration abilities of the algorithm. Each time an ant decides to move from feature  $i$  to the next feature, it generates an accidental numerical  $q$ . If  $q > q_0$ , each feature can be chosen proportionate to its possibility using Eq. (3) (Exploration). Otherwise, the most useful feature until now will be chosen to utilize Eq. (4) (exploration). The state transition rule is composed of Eqs. (3) and (4) and are called the

**Table 1**

Datasets and their characteristics.

Dataset	Feature value	# Instances	# Features	# Classes	Source
1 Arcene	Continuous	200	10,000	2	(Li et al., 2017)
2 Clean1	Discrete	476	167	2	(Deng, Li, Wang, & Wan, 2021)
3 Gravier	Continuous	168	2905	2	(Roughened Random Forests for Binary Classification - ProQuest. (n.d.))
4 Jaffe	Discrete	213	676	10	(JAFFE Dataset   Papers With Code. n.d.)
5 Khan	Continuous	63	2308	4	(Khan et al., 2001)
6 Leukemia	Discrete	72	7070	2	(Li et al., 2017)
7 Lung	Continuous	203	3312	5	(Li et al., 2017)
8 lymphoma	Discrete	96	4026	9	(Li et al., 2017)
9 Madelon	Discrete	2000	500	2	(Deng et al., 2021)
10 Musk2	Discrete	6598	166	2	(UCI Machine Learning Repository: Musk (Version 2) Data Set, n.d.)
11 NCI60	Continuous	64	6830	14	(NCI60: NCI 60 Data in ISLR: Data for an Introduction to Statistical Learning with Applications in R, n.d.)
12 Prostate-GE	Continuous	102	5966	2	(Li et al., 2017)
13 Sorlie	Continuous	85	456	5	<a href="https://search.r-project.org/CRAN">https://search.r-project.org/CRAN</a>
14 SU	Continuous	102	5565	4	(Li et al., 2017)

**Table 2**  
Parameters setting of SemiACO.

Parameter	Explanation	Quantity
$\alpha$	The Learning rate of TD	0.5
$\gamma$	The Discount rate of TD	0.8
$\rho$	Pheromone decay rate	0.15
NumCycle	The numeral of times the algorithm should repeat	30
NumAnt	The count of ants that search in the features space	80
NumF	The count of features every ant should cross	$1/8d \leq \text{NumF} \leq 1/6d$
$q$	exploration-exploitation coefficient	Equation (9) (power = 0.7)
$\beta$	The exchange between heuristic information and pheromones	1
$m$	The count of most significant features that choose	$10 \leq m \leq 100$

“pseudo-random proportional” rule.

After ants traversed some features on the graph, the pheromone paths were updated by every ant separately. The global updating rule updates the pheromone according to the following equation (Paniri et al., 2020):

$$\tau_i(t+1) = (1 - \rho)\tau_i(t) + \Delta\tau_i, \quad (5)$$

where  $\tau_i(t)$ ,  $\tau_i(t+1)$  are the pheromone values of feature  $i$  at times  $t$  and  $t+1$  separately, and  $\rho$  is a pheromone decay rate parameter.  $\Delta\tau_i$  represents the pheromone increment value for feature  $i$  based on some criteria.

Ultimately, a feature with a high quantity of pheromones is a profitable feature. Thus, the optimal feature subset is specified by choosing  $m$  ( $m > d$ ) top features determined by the amount of pheromone.

### 3. Proposed algorithm

In this part, we will discuss the proposed semi-supervised feature selection algorithm using ant colony optimization. Here, graph nodes demonstrate the features, and edges display the relationship among features. The proposed algorithm does not contain any learning algorithms and is a filter-based technique. In the following parts, we will discuss the proposed SemiACO characteristics that learn with nonlinear heuristic learning instead of a linear one. The general structure of the recommended algorithm shows in Fig. 3.

#### 3.1. Pheromone information

The pheromone sequence is a principal part of the ACO algorithm to build a good solution. In the proposed algorithm, every ant passes through the graph nodes (features) and puts some pheromone on them, forming a  $d$ -dimensional vector  $\tau$  named the pheromone sequence. Pheromone initialization affects ACO performance; the initialization of pheromones with a constant value slows down the ACO convergence speed. Therefore, to increase the convergence speed and improve the efficiency, rather than initializing the pheromone with a constant value, the mutual information among the features and the class label is calculated for pheromone initialization. Due to the semi-supervised nature of the proposed method, the mutual information calculation is conducted based on just labeled instances. The mutual information

**Table 3**

The obtained p-values by the Friedman test for the accuracy metric based on 20 percent labeled data.

dataset	SemiACO against					ENCLS	Self	TRCFS
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF			
Arcene	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Clean1	0.0005(+)	0.0018(+)	0.0833(=)	0.0005(+)	0.0005(+)	0.0377(+)	0.0005(+)	0.0005(+)
Gravier	0.0047(+)	0.0047(+)	0.0004(+)	0.7237(=)	0.0015(+)	0.0047(+)	0.0339(+)	0.7237(=)
Jaffe	0.0056(+)	0.0005(+)	0.2207(=)	0.0018(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0005(+)
Khan	0.0005(+)	0.0005(+)	0.0005(+)	0.4884(=)	0.0005(+)	0.0005(+)	0.2987(=)	0.0005(+)
Leukemia	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0051(+)
Lung	0.0005(+)	0.0143(+)	0.7264(=)	0.0005(+)	0.0005(+)	0.0005(+)	0.7264(=)	0.0802(=)
Lymphoma	0.0005(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)	0.0018(+)
Madelon	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.4884(=)	0.0005(+)	0.0005(+)
Musk2	0.0056(+)	0.0005(+)	0.1659(=)	1.0000(=)	0.0005(+)	0.0005(+)	0.0005(+)	1.0000(=)
NCI60	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Prostate-GE	0.0005(+)	0.0377(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.1659(=)
Sorlie	0.0018(+)	0.0018(+)	0.0005(+)	0.0018(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0018(+)
SU	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0016(+)	0.0005(+)

**Table 4**

The obtained p-values by the Friedman test for the F-score metric based on 20 percent labeled data.

dataset	SemiACO against							
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Clean1	0.0018(+)	0.0018(+)	0.0377(+)	0.0005(+)	0.0005(+)	0.0833(=)	0.0018(+)	0.0005(+)
Gravier	0.0833(=)	0.0377(+)	0.0018(+)	0.4884(=)	0.0377(+)	0.2987(=)	0.0377(+)	0.2987(=)
Jaffe	0.0056(+)	0.0005(+)	0.4884(=)	0.0018(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Khan	0.0005(+)	0.0005(+)	0.0005(+)	1.0000(=)	0.0005(+)	0.0005(+)	0.4884(=)	0.0005(+)
Leukemia	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0056(+)
Lung	0.0377(+)	0.0377(+)	0.4884(=)	0.0005(+)	0.0005(+)	0.0005(+)	1.0000(=)	0.0377(+)
Lymphoma	0.0018(+)	0.0018(+)	0.0018(+)	0.0005(+)	0.0018(+)	0.0005(+)	0.0018(+)	0.1659(=)
Madelon	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.4884(=)	0.0005(+)	0.0005(+)
Musk2	0.7290(=)	0.2987(=)	1.0000(=)	1.0000(=)	0.0056(+)	0.0153(+)	0.0018(+)	1.0000(=)
NCI60	0.0005(+)	0.0005(+)	0.0005(+)	0.0153(+)	0.0005(+)	0.0005(+)	0.0056(+)	0.0005(+)
Prostate-GE	0.0005(+)	0.0153(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.1659(=)
Sorlie	0.0018(+)	0.0018(+)	0.0005(+)	0.0018(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0018(+)
SU	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)	0.0005(+)

between two variables  $x$  and  $y$  is formulated as follows (Bennasar, Hicks, & Setchi, 2015):

$$mi(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log(p(x)p(y)/p(x, y)). \quad (6)$$

The higher the mutual information value of the two variables, the more the two variables are similar.

### 3.2. Solution construction

In the proposed algorithm, the ants accidentally select the features initially. Every ant creates a solution utilizing the laws of greed and probability while visiting various features in the search space. State transfer aims to find the optimal features with the highest number of pheromones and gain heuristic information about the movement of ants. In other words, the transition purpose is to balance the state of balance between exploration and exploitation abilities with the application of heuristic and pheromone information. At each stage, the ant applies the following characteristic by using the law of choice of probable or greedy action (Paniri et al., 2021):

$$p_i^k(t) = \begin{cases} [\tau_i(t)] [V_i(t)]^\beta / \sum_{u \in N^k} [\tau_u(t)] [V_u(t)]^\beta, & \forall i \in N^k, \text{ if } q > q_0 \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

$$i = \arg \max_{u \in N^k} \left\{ [\tau_u] [V_u]^\beta \right\}, \text{ if } q \leq q_0, \quad (8)$$

where  $N^k$  is a set of unseen features that ant  $k$  can visit;  $\tau_i$  is the value of the pheromone associated with a feature  $i$ , and  $V_i$  is the degree of the advisability of exploration, which is a function of the value-state associated with the feature  $i$  (Benabdeslem & Hindawi, 2011). The value of parameter  $\beta$  differs in ranges  $[0, 1]$  and controls the exchange among the amount of pheromone and the heuristic information. If  $\beta = 0$ , heuristic information is ignored, and the decision is based on the former action history. The parameter  $q$  is a random variable that is uniformly dispensed in ranges  $[0, 1]$ , and  $q_0 \in [0, 1]$  is a constant numeral. These two parameters determine the comparative significance of exploitation and exploration abilities. If  $q > q_0$ , every feature has an equal opportunity of being seen, depending on the ant's choice, indicating exploration. And if  $q < q_0$ , then each ant sees the most useful feature according to the most significant quantity of pheromone and its exploratory worth, which leads to more exploitation.

The critical point is that the exchange between exploration and extraction must be controlled. Here we set the exploration extraction factor  $q$  by exerting the multinomial decay rate, considering the number of iterations. The ants explore more in the initial repetitions and have more exploration at the end of the iterations. The mathematical form of the multinomial decay rate is as follows (Paniri et al., 2021):

$$q = (\text{start}_{\text{rate}} - \text{end}_{\text{rate}}) \times (1 - k/\text{NumF})^{\text{power}} + \text{end}_{\text{rate}}. \quad (9)$$

Here,  $\text{start}_{\text{rate}}$  and  $\text{end}_{\text{rate}}$  set the lower limits, and the power of the decay rate, respectively.  $\text{NumF}$  represents the number of features every ant must visit in every iteration, and parameter  $k$  is the current number of iterations.

### 3.3. Heuristic learning

ACO algorithms generally utilize a linear heuristic function, which ignores other information, including nonlinear relationships. Therefore, the proposed technique utilizes the nonlinear heuristic learning function rather than the linear function. MIC indicates the nonlinear and linear dependence between two variable pairs. The MIC value is in ranges  $[0, 1]$ , where 0 represents independence, and 1 indicates the maximum relationship between variables (Cao, Chen, Chen, Zhang, & Yuan, 2021). MIC is looking for a smaller subset of the original feature set with the most relationship and has two excellent characteristics: generality and equality. It can identify various relevancy, including linear and nonlinear, functional and non-functional. Through empirical comparison, MIC demonstrates good fairness and versatility (Pan, 2021).

The reward function consists of two unsupervised and supervised metrics. The first is a redundancy matrix with dimensions  $d \times d$  is called  $fclcos$ . The cosine coefficient formula to calculate the redundancy between features is as follows (Lee Rodgers & Alan Nice Wandering, 1988):

$$\text{Cosine}(X, Y) = \left| \sum_{i=1}^n (X_i Y_i) / \left( \sqrt{\sum_{i=1}^n X_i^2} \right) \left( \sqrt{\sum_{i=1}^n Y_i^2} \right) \right|, \quad (10)$$

where  $X$  and  $Y$  are two  $n$ -dimensional feature vectors, and  $n$  is the number of samples. Since the algorithm is proposed for semi-supervised data, we utilize the cosine similarity to compute the redundancy between all samples (both labeled and unlabeled).

The second metric is a relevancy vector with size  $d$  called  $flcorr$ . MIC is utilized to calculate the relevancy between features and the class label considering labeled samples, which is as follows (Zhu, You, & Liu, 2019):

$$\text{MIC}(X, Y) = \frac{\max}{|X||Y|T} MI(X, Y) / \log_2(\min(|X|, |Y|)), \quad (11)$$

where  $X$  and  $Y$  are two  $n$ -dimensional features vector,  $|X|$  and  $|Y|$  are the number of bins on the  $X$ -axis and  $Y$ -axis, respectively,  $T$  is the upper limit of the grid partition of  $X \times Y$  that increases with the number of instances included in the dataset,  $MI(X, Y)$  demonstrates the mutual information  $X, Y$ .

The formula for calculating the entropy of one variable and two variables is as follows (Zhu et al., 2019):

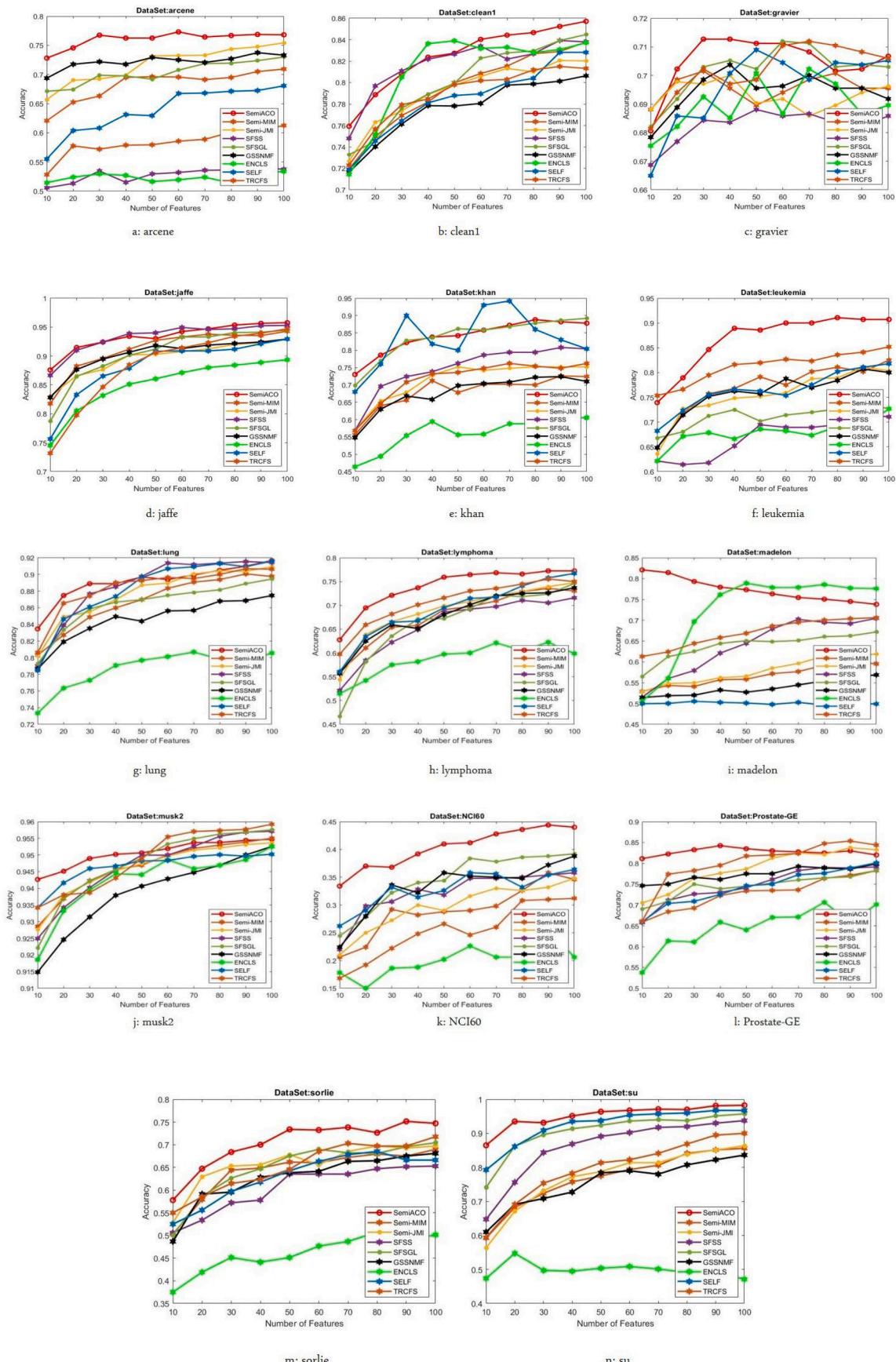
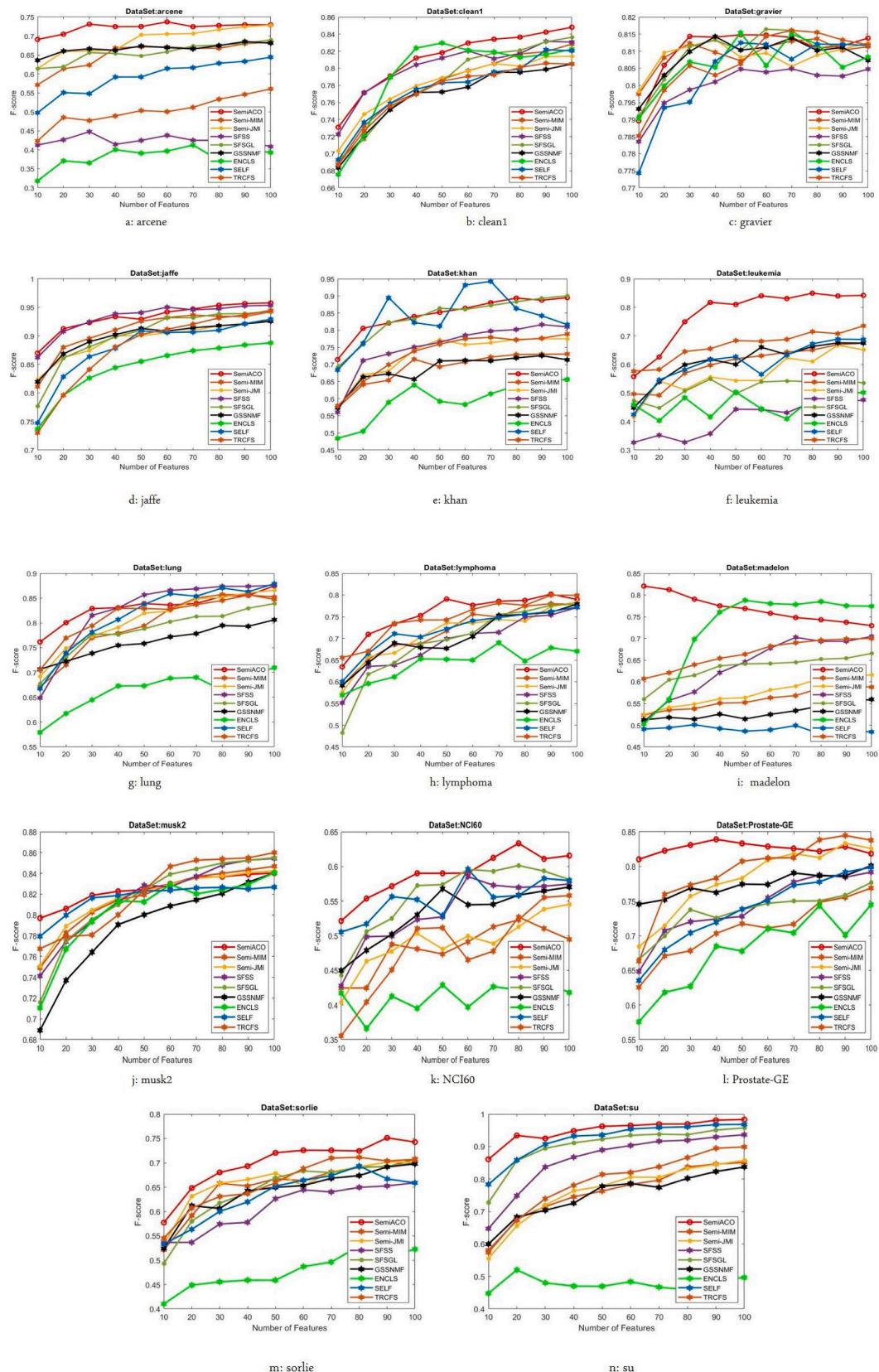
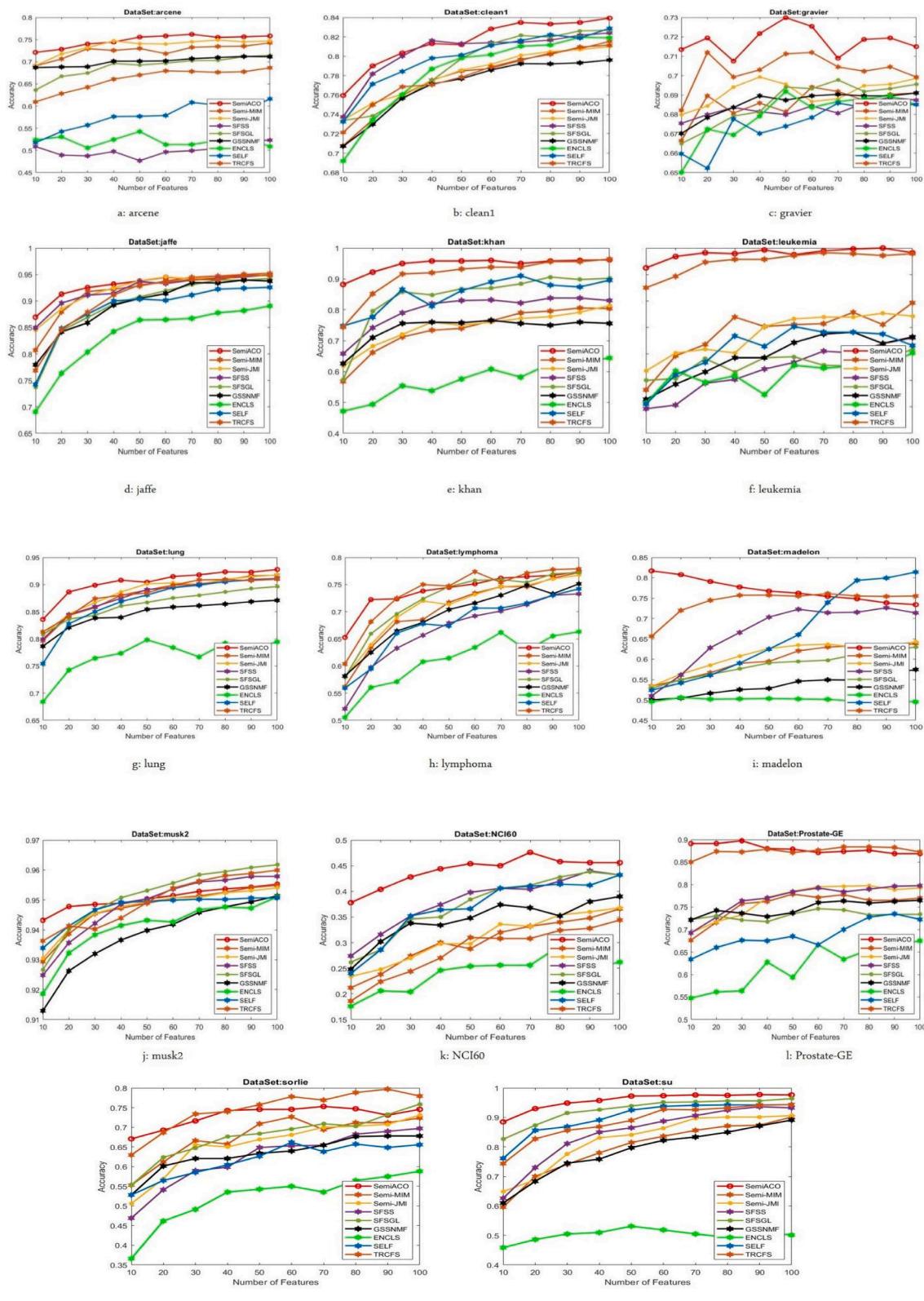


Fig. 4. The classification results based on Accuracy measure for 20 percent labeled data.



**Fig. 5.** The classification results based on the F-score measure for 20 percent labeled data.



**Fig. 6.** The classification results based on the Accuracy measure for 40 percent labeled data.

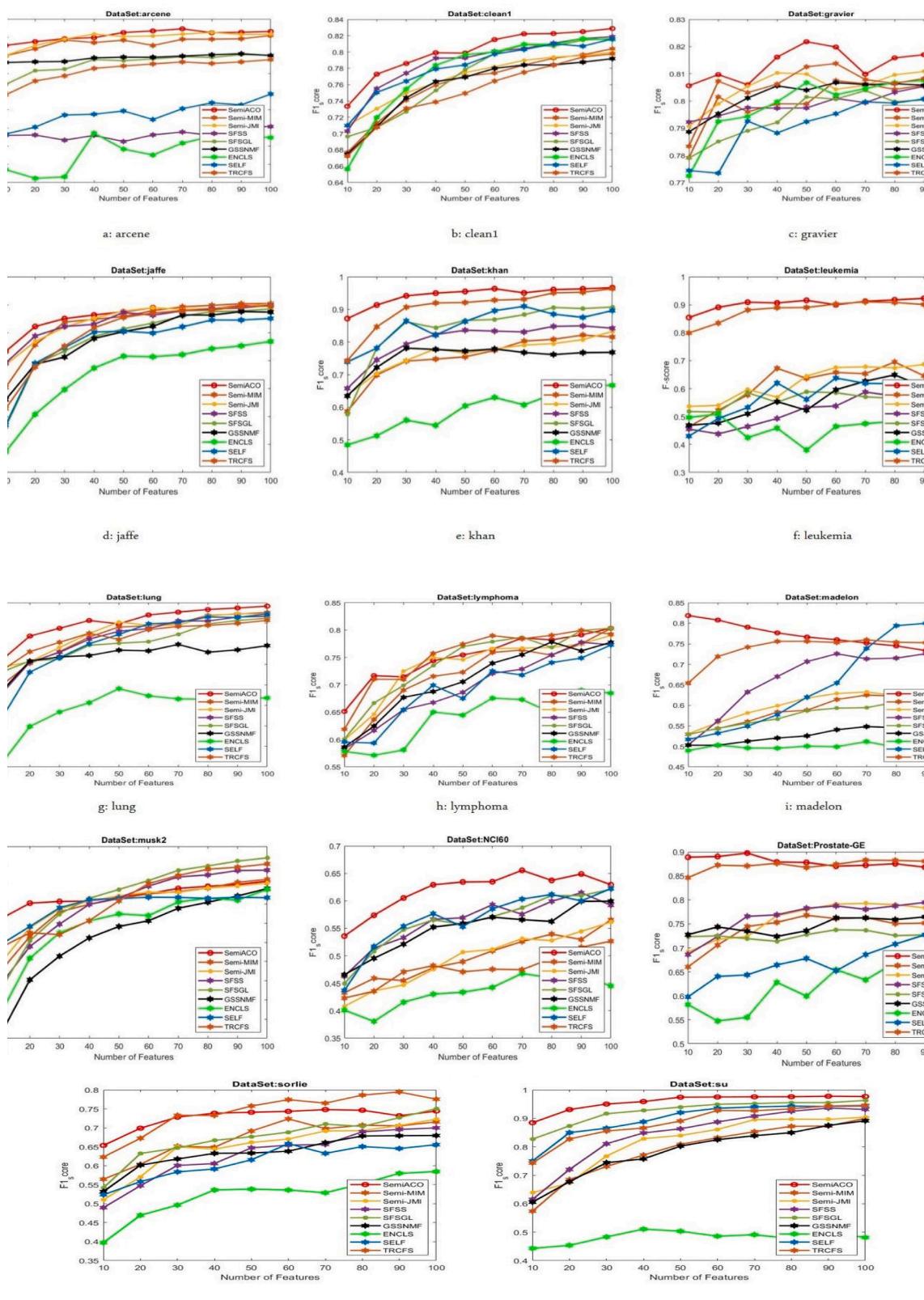
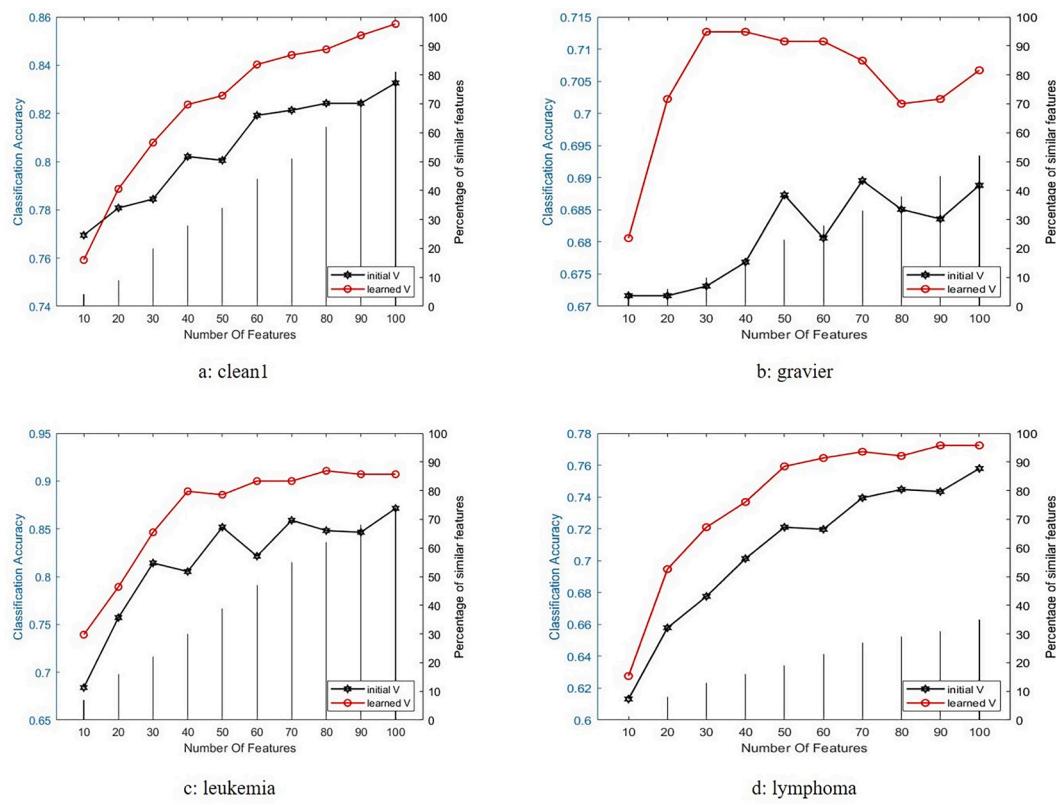


Fig. 7. The classification results based on the F-score measure for 40 percent labeled data.



**Fig. 8.** The effect of heuristic learning on V state-value before and after the algorithm's execution.

**Table 5**

The obtained p-values by the Friedman test for the accuracy metric based on 40 percent labeled data.

dataset	SemiACO against							
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Clean1	0.0005(+)	0.0005(+)	0.0056(=)	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)	0.0005(+)
Gravier	0.0015(+)	0.0015(+)	0.0004(=)	0.0004(+)	0.0004(+)	0.0004(+)	0.0004(+)	0.4795(=)
Jaffe	0.0087(+)	0.0833(=)	0.0153(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0056(=)
Khan	0.0005(+)	0.0005(+)	0.0018(+)	0.0153(+)	0.0005(+)	0.0005(+)	0.2987(=)	0.0018(+)
Leukemia	0.0016(+)	0.0015(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Lung	0.7264(=)	0.7264(=)	0.0802(=)	0.0016(+)	0.0005(+)	0.0005(+)	0.0358(+)	0.2938(=)
Lymphoma	0.0051(+)	0.0018(+)	0.0005(=)	0.0153(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.4884(=)
Madelon	0.0005(+)	0.0005(+)	0.0005(=)	0.0005(+)	0.0005(+)	0.0005(+)	0.1659(=)	0.4884(=)
Musk2	0.0056(+)	0.0018(+)	1.0000(=)	0.1659(=)	0.0005(+)	0.0005(+)	0.0018(+)	1.0000(=)
NCI60	0.0005(+)	0.0005(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Prostate-GE	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Sorlie	0.0018(+)	0.0005(+)	0.0005(+)	0.0153(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0018(-)
SU	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)

$$H(X) = - \sum_i p(X_i) \log(p(X_i)). \quad (12)$$

$$H(X, Y) = - \sum_i \sum_j p(X_i, Y_i) \log(p(X_i, Y_i)). \quad (13)$$

For cosine similarity and MIC, if the value is closer to 0, two variables are independent, and if it is nearer to 1, they are dependent.

From Eq. (2), when the agent goes from state  $s_t$  to state  $s_{t+1}$  by taking action  $a$ , the rewards of  $r_{t+1}$  can be calculated as follows:

$$r_{t+1} = flCorr_{s_{t+1}}, l/1 + flCos_{s_t+s_{t+1}}, \quad (14)$$

where  $flCorr_{s_{t+1}}$ ,  $l$  is the feature-label relevancy value of the subsequent feature  $s_{t+1}$ , and all its related labels, and  $flCos_{s_t+s_{t+1}}$  is the feature relevancy between the present feature  $s_t$  and the subsequent feature  $s_{t+1}$ .

This combination offers more rewards to features with high relevancy with the class label simultaneously and has the least redundancy with the earlier chosen feature. In every iteration, the ants update the  $V$ -value vector locally as they build their solution. At the end of the repetition, vector  $V$  of every ant  $k$  is updated globally by averaging the whole:

$$V_{s_t} = 1/NumAnt \sum_{k=1}^{NumAnt} V_{s_t}^k, \quad (15)$$

The  $NumAnt$  parameter is the count of ants making the solution and choosing the subsets.

### 3.4. Pheromone update

The pheromone update with determining a  $d$ -dimensional vector named feature counter increases when every ant visits a feature. Once the ants have completed their solutions, the following global pheromone

**Table 6**

The obtained p-values by the Friedman test for the F-score metric based on 40 percent labeled data.

dataset	SemiACO against					ENCLS	Self	TRCFS
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF			
Arcene	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Clean1	0.0005(+)	0.0005(+)	0.0056(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)	0.0005(+)
Gravier	0.0018(+)	0.0153(+)	0.0056(+)	0.0005(+)	0.0018(+)	0.0018(+)	0.0005(+)	0.0056(+)
Jaffe	0.0377(+)	0.0833(+)	0.0153(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0018(+)
Khan	0.0005(+)	0.0005(+)	0.0056(+)	0.0377(+)	0.0018(+)	0.0005(+)	0.4884(=)	0.0018(+)
Leukemia	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Lung	1.0000(=)	0.7290(=)	0.2987(=)	0.0153(+)	0.0005(+)	0.0005(+)	0.2987(=)	0.0377(+)
Lymphoma	0.0153(+)	0.0153(+)	0.0005(+)	0.2987(=)	0.0005(+)	0.0005(+)	0.0005(+)	0.7290(=)
Madelon	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.1659(=)	0.4884(=)
Musk2	0.7290(=)	0.4884(=)	0.4884(=)	0.1659(=)	0.0005(+)	0.0005(+)	0.0377(+)	1.0000(=)
NCI60	0.0005(+)	0.0005(+)	0.0153(+)	0.0056(+)	0.0005(+)	0.0005(+)	0.0153(+)	0.0005(+)
Prostate-GE	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)
Sorlie	0.0056(+)	0.0005(+)	0.0005(+)	0.0056(+)	0.0018(+)	0.0005(+)	0.0005(+)	0.0018(-)
SU	0.0005(+)	0.0005(+)	0.0005(+)	0.0056(+)	0.0005(+)	0.0005(+)	0.0005(+)	0.0005(+)

**Table 7**

The overall win/tie/loss based on the Friedman test.

Dataset	SemiACO against					ENCLS	Self	TRCFS
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF			
Accuracy on 20 percent of labeled data	14/0/0	14/0/0	10/4/0	11/3/0	14/0/0	13/1/0	12/2/0	10/4/0
F-measure on 20 percent of labeled data	12/2/0	13/1/0	11/3/0	12/2/0	14/0/0	11/3/0	12/2/0	10/4/0
Accuracy on 40 percent of labeled data	13/1/0	12/2/0	12/2/0	13/1/0	14/0/0	14/0/0	12/2/0	8/5/1
F-measure on 40 percent of labeled data	12/2/0	12/2/0	12/2/0	12/2/0	14/0/0	14/0/0	11/3/0	10/3/1
Overall	51/5/0	51/5/0	45/11/0	48/8/0	56/0/0	52/4/0	47/9/0	38/16/2

update rule applies to update the pheromone value for every feature  $i$ :

$$\tau_i(t+1) = (1 - \rho) \tau_i(t) + AntFC(i) / \sum_{j=1}^d AntFC(j). \quad (16)$$

Algorithm 2 shows the summary of the proposed method, i.e., semi-supervised feature selection based on the SemiACO algorithm.

## Algorithm 2. Semi-supervised feature selection based on the SemiACO algorithm.

Inputs:  $X_L$  = labeled samples,  $X = X_L \cup X_U$  (all samples labeled and unlabeled),  $Y_L$  = class label of the labeled samples,  $d$  features  $NumCycle$ : The maximum number of iterations in ACO.  $NumAnt$ : The number of ants.  $NumF$ : number of features that every ant should be chosen in every cycle.

Output: Selected feature subset with the size of  $m$ Begin

1. Compute  $d \times d$  redundancy matrix  $fICos_{ij}$  between all features  $X$  (labeled and unlabeled samples) by Eq. (10),  $\forall i, j = 1, \dots, d$ .
2. Compute  $d$ -dimensional relevancy vector  $fICorr_{ij}$  between features and class label of labeled samples  $X_L$  by Eq. (11),  $\forall i = 1, \dots, d$
3. Initialize the pheromone vector,  $\tau_i = MI(F_i, Y_L)$  accordance with Eq. (6)  $\forall i = 1, \dots, d$
4. Initialize the state-value vector  $V_i = MIC(F_i, Y_L)$  accordance with Eq. (11)  $\forall i = 1, \dots, d$
5. For  $c = 1$  to  $NumCycle$  do
6. Generate and place ants randomly on the features
7. Initialize ant feature counter for every feature,  $AntFC[i] = 0, \forall i = 1, \dots, d$
8. For  $ant = 1$  to  $NumAnt$  do
9.  $NumF$  = generate a random integer in ranges  $[d/8, d/6]$
10. For  $k = 1$  to  $NumF$  do (heuristic learning)
  - a.  $a$  = Select the following feature  $f$  by the state transition rule according to Eqs. (7 and 8)
  - b. Take action  $a$  observe the reward  $r_{t+1}$  (Eq. 14), and next state  $s_{t+1}$
  - c. Update state-value vector  $V$  by every ant locally by Eq. (2)
  - d. Increment the  $AntFC$  for the seen feature  $f$ ,  $AntFC[f] = AFC[f] + 1$
11. Endfor
12. Endfor
13. Update state-value vector  $V$  by using Eq. (15)
14. Update the pheromone vector by applying for the global update role by Eq. (16)
15. Endfor
16. Sort features by their associated pheromone trail worth in descending order
17. Choose top  $m$  features as output

End

**4. Experiments setting**

This part describes the simulation environment, datasets, classifiers, evaluation criteria, and parameter settings. All experiments were implemented utilizing matlab\_r2021a on an Intel Core i7 CPU with 16 GB of RAM in the Microsoft Windows 10 operating system. We repeated every experiment through 30 independent runs for every dataset and algorithm to gain stable results. In every run, by utilizing the hold-out cross-validation, 60 % of the dataset samples were randomly chosen as the training collection and the remaining 40 % as the test set. Due to the semi-supervised nature of the proposed algorithm, experiments were done with random labeling on 20 % and 40 % of the training samples separately.

**4.1. Datasets**

We performed the experiments based on fourteen datasets: Arcene, Clean1, Gravier, Jaffe, Khan, Leukemia, Lung, lymphoma, Madelon, Musk2, NCI60, Prostate-GE, Sorlie, and Su. The characteristics of the datasets are listed in Table 1.

**4.2. Classifier**

We used the K-Nearest Neighbor classifier (KNN) to assess the categorization of our proposed algorithm (Liao & Vemuri, 2002). The  $k$  parameter utilized in the KNN classifier is set to 5. A different number of features are selected in the range {10, 20, 30, 40, 50, 60, 70, 80, 90, 100} to assess the performance of the semi-supervised feature selection procedures.

**4.3. Performance assessment criteria**

To assess the efficiency of the proposed algorithm and competing techniques, we utilized accuracy and F-score metrics (Hashemi et al., 2021e). First, observe the following notions:

**Table 8**  
Run-time comparisons on 20 percent labeled data.

SemiACO	217.538	0.1271	17.1144	0.8015	10.3242	217.9848	26.1468	68.9694	0.9635	0.2803	106.9454	90.6152	0.8879	103.4424
Semi-MM	1.01E + 04	2.1964	855.1529	25.8569	639.7204	5.70E + 03	1.16E + 03	1.72E + 03	126.9358	30.7604	4.54E + 03	4.83E + 03	23.1184	3.45E + 03
Semi-JMI	1.01E + 04	2.1964	855.1528	25.8568	639.7204	5.70E + 03	1.16E + 03	1.72E + 03	126.9358	30.7603	4.54E + 03	4.83E + 03	23.1184	3.45E + 03
SFSS	39.232	0.017	6.2018	0.069	2.0407	39.9451	7.0634	8.1868	0.8224	21.4537	49.1071	38.2078	0.0599	23.9235
SFSGL	1.06E + 03	0.0336	36.5488	0.1391	24.3326	841.3363	102.8575	197.4651	7.6871	57.1856	775.7263	810.2006	0.5128	737.1245
GSSNMF	0.2576	0.0144	0.0842	0.0276	0.0322	0.0921	0.143	0.1203	0.9121	2.705	0.0958	0.1142	0.0174	0.1518
ENCLS	10.8828	32.2177	8.2375	6.5901	2.0665	2.8181	13.2009	4.1747	283.8907	2.83E + 03	2.527	4.599	2.7092	4.9766
SELF	223.5991	0.1467	45.4768	0.7157	33.8233	348.3809	45.3454	294.9491	1.5432	4.5832	309.5145	197.2423	1.9155	408.4717
TRCFS	2.4106	0.0147	0.2191	0.0212	0.0875	0.9754	0.2826	0.3269	1.024	8.8886	0.9253	0.864	0.0108	0.6033
Arcene	Clean1	Gravier	Jaffe	Khan	Leukemia	Lung	lymphoma	Madelon	Musk2	NCI60	Prostate-GE	Sofie	su	

**Table 9**  
Run-time comparisons on 40 percent labeled data.

SemiACO	404.1359	1.3998	56.3883	3.4623	38.4948	185.2011	94.9232	61.0064	11.2559	738.4742	156.5278	2.247	133.8198	
Semi-MM	865.8016	0.854	449.2347	4.1755	62.3079	308.8422	204.6005	96.9268	9.5372	2.1374	549.2769	288.714	2.4466	200.26
Semi-JMI	865.8015	0.8453	449.2345	4.1754	62.3078	308.8415	204.6004	96.9267	9.5372	2.1372	549.2767	288.7139	2.4465	200.26
SFSS	39.9798	0.0501	9.9898	0.1243	4.2744	41.5873	13.4017	8.5217	0.7759	3.3136	73.6994	42.6458	0.0785	27.4735
SFSGL	1.45E + 03	0.1266	333.0907	0.173	62.5907	1.44E + 04	160.2422	1.82E + 03	5.7256	54.6615	1.10E + 03	1.39E + 03	4.018	1.15E + 04
GSSNMF	0.2794	0.052	0.1279	0.0588	0.0452	0.1263	0.2615	0.0741	0.6558	2.7135	0.1251	0.1098	0.0209	0.0988
ENCLS	13.2143	134.673	16.8042	15.1251	4.0143	2.523	25.7691	2.912	2.17E + 03	2.82E + 04	3.6843	5.5476	4.8377	3.7224
SELF	505.1395	0.444	76.2996	1.7457	74.6716	757.9332	240.2031	164.0963	1.6739	5.0627	486.4758	285.9702	4.5501	288.4867
TRCFS	2.343	0.0465	0.3746	0.0363	0.1437	0.8355	0.5089	0.3652	0.9036	3.7171	0.8981	0.736	0.0119	0.6594
Arcene	Clean1	Gravier	Jaffe	Khan	Leukemia	Lung	lymphoma	Madelon	Musk2	NCI60	Prostate-GE	Sofie	su	

**Table 10**

The computational complexity of SemiACO algorithm.

Step	Computational Complexity (Big Omicron)
Step 1 (Redundancy matrix)	$O(nd^2)$
Step 2 (Relevancy vector)	$O(d^2)$
Step 3 (Pheromone vector)	$O(d^2)$
Step 4 (State-value vector)	$O(d^2)$
Steps 5–15 (ACO process)	$O(NumCycle \times NumAnt \times d^2)$
Step 16 (Sorting the features)	$O(d\log d)$
Overall	$O(nd^2 + d^2 + NumCycle \times NumAnt \times d^2)$
Description of symbols	$d$ —number of features, $n$ —number of instances, $l$ —number of labeled samples, $NumCycle$ —number of iterations, $NumAnts$ —number of ants.

- True positive (*TP*): percentage of positive instances rightly categorized.
- False-positive (*FP*): percentage of negative instances mistakenly categorized as positive.
- True negative (*TN*): percentage of negative instances rightly categorized.
- False-negative (*FN*): percentage of positive instances mistakenly categorized as negative.

Accuracy and F-score criteria are expressed as follows:

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN). \quad (17)$$

$$F\text{-score} = TP / (TP + 1/2(FP + FN)). \quad (18)$$

#### 4.4. Parameter settings

The swarm intelligence algorithms need to set parameters that are specified experimentally. These parameters are tuned experimentally. These parameters' values are listed in Table 2. The number of ants is set to 80 ( $NumAnt = 80$ ), the pheromone decay rate is set to 0.15 ( $\rho = 0.15$ ), and the number of repetitions that ants should repeat is set to 30 ( $NumCycle = 30$ ). The number of features that every ant can visit in every iteration is set to a random number between one-eighth and one-sixth of the primary features ( $1/8d \leq NumF \leq 1/6d$ ). The learning rate and discount rate of the TD heuristic learning are set to 0.5 and 0.8 ( $\alpha = 0.5, \gamma = 0.8$ ). The *start\_rate*, *end\_rate*, and the power parameters of the multinomial decay rate are set to 1, 0, and 0.7, and parameter  $\beta$  is set to 1 ( $\beta = 1$ ). The final number of the top feature that should be chosen ( $m$ ) is a numeral between 10 and 100, but feel free to set it to every value up to the main feature number  $1 \leq m \leq d$ . The suggested parameter's values are considered for all competitive methods based on their corresponding article.

## 5. Results and discussion

In this part, we compare the performance of the proposed algorithm SemiACO against multiple algorithms in the literature: SFSS (Ma et al., 2011), SELF (Sugiyama et al., 2010), EnsCLS (Hindawi, Elghazel, & Benabdeslem, 2013), SFSGL (Sheikhpour et al., 2018), GSSNMF (Jia et al., 2020), semiJMI and SemiMIM (Sechidis & Brown, 2018) and TRCFS (Liu et al., 2010).

Tables 3 and 4 display the execution time of the algorithms in cases 20 % and 40 % of the samples are randomly labeled. Figs. 4, 5, 6, and 7 indicate the accuracy and F-score of algorithms. Fig. 8 indicates the effect of heuristic learning on the *V* state value before and after the execution of the SemiACO.

### 5.1. Comparison with other semi-supervised feature selection techniques

According to the observation and outcomes, it can be seen that the SemiACO algorithm can improve the classification performance compared to other semi-supervised feature selection techniques in the literature.

In Fig. 4 to Fig. 8, the classification performance of the SemiACO against competitive methods is presented based on two different splits of data. In all these figures, the red diagram shows the performance of the proposed method. Also, the number of selected features and the classification metric's value are presented in the vertical and horizontal axes, respectively.

Fig. 4 displays the average classification accuracy of the SemiACO and competitive methods when 20 % of the training samples are labeled. It can be seen that the proposed algorithm has achieved better classification accuracy in all cases compared to other techniques, especially for Arcene, Clean1, Gravier, Leukemia, Lymphoma, Madelon, NCI60, Prostate-GE, Sorlie, and SU datasets.

Fig. 5 demonstrates the average F-score of the SemiACO when 20 % of the training samples are labeled. It can be seen that the proposed algorithm has achieved better F-score value in all cases compared to other techniques, especially for Arcene, Clean1, Leukemia, Lymphoma, Madelon, NCI60, Prostate-GE, Sorlie, and SU datasets.

Fig. 6 displays the average accuracy of the SemiACO when 40 % of the training samples are labeled. The results of 40 % labeling of training samples show, like the result of 20 % labeling, SemiACO has shown better classification accuracy in all cases compared to other techniques, especially for Arcene, Clean1, Gravier, leukemia, Lymphoma, Madelon, NCI60, Prostate-GE, Sorlie, khan, and SU datasets. In all datasets, SemiACO has maintained its high accuracy in both 20 % and 40 % labeling of training samples compared to algorithms.

Fig. 7 displays the F-score of the SemiACO when 40 % of the training samples are labeled. The results of 40 % labeling of training samples show, like the result of 20 % labeling, SemiACO has shown better F-score value in all cases compared to other techniques, especially for Arcene, Clean1, Gravier, Leukemia, Lymphoma, Madelon, NCI60, Prostate-GE, Sorlie, Khan, and SU datasets.

A nonparametric Friedman test (Friedman, 1940) is conducted to compare the significance of the results based on a statistical approach. The significant level for this statistical test is considered 0.05. These comparisons are reported in Tables 3 to 6, and Table 7 presents the overall win/tie/loss. In Tables 3 to 6, the (+) sign means that the proposed SemiACO is statistically better than other methods. Also, (=) and (-) signs show equal and worst performance, respectively. Tables 8 and 9 demonstrate the execution times (in seconds) when 20 percent and 40 percent of data samples are labeled.

### 5.2. Discussion

This paper uses the ACO algorithm to investigate the best feature subset in semi-supervised data samples. Compared to other metaheuristic algorithms, ACO successfully solved combinatorial optimization problems, mainly feature selection problems. Compared to other presented methods of semi-supervised feature selection and other metaheuristic algorithms, the main difference between this method is the heuristic selection. In the mentioned methods, the feature selection process is based on a specific heuristic, which does not change until the end of the algorithm execution. However, in the proposed method, a dynamic heuristic selection process is considered and updated with each iteration of the algorithm. We have conducted experiments to demonstrate the effect of heuristic learning using TD reinforcement learning. Fig. 8 displays the results plotted on the left and right y-axes, where the right side shows the percentage of top similar features selected from initial *V* and Learn-*V*. The left side compares the classification accuracy of the state value heuristic function *V* before and after the algorithm's execution. There are interpretations of the numbers:

- As we mentioned in the proposed method section, we initialized the value of V with MIC. Therefore, MIC provides the base knowledge to TD. The figures demonstrate how heuristic learning significantly improves the quality of the selected features before and after running the algorithm. During the algorithm execution, these improvements lead the ants to choose the most promising features that result from using heuristic learning instead of using a fixed heuristic function such as MIC.
- A good feature selection method gets higher classification performance with the lowest number of features. The results show that when the number of features is low, around 30 features, the classification accuracy of the learned-V is better than the initial-V. Furthermore, when the number of features increases, the classification accuracy of the learned-V is higher than the initial-V looks approximately. It is clear from the bar charts that when the number of features increases, the similar features increase too, and this results in an approximate equal performance between the learned-V and the initial-V features.

Based on the observations and results, we can conclude that the semi-supervised feature selection using Ant-TD can improve the classification performance of semi-supervised learning by comparing existing methods in the literature. There is some interpretation of the performance of the proposed method:

- The first characteristic of the proposed method is the first time a semi-supervised feature selection method has been proposed based on ACO. Since ACO has not been used in semi-supervised learning, we decided to use ACO in semi-supervised feature selection due to its high efficiency and power.
- The second characteristic of the new approach is that the SemiACO initializes the value function with a nonlinear heuristic function instead of a linear one. Using a nonlinear function instead of a linear one improves the learning performance and increases the algorithm's accuracy.

A statistical test is used based on the Friedman test to validate the results. The overall amount of win/tie/loss by the statistical comparisons are presented in [Table 9](#). We can witness the superiority of the proposed method against the other algorithms in two classification metrics and two different semi-supervised perspectives. In most cases, the proposed method outperforms the competitive methods statistically (388 cases among 448 cases). In two cases, another method performs better, and the performance is a tie in the remainder of 58 cases.

Also, in [Tables 8 and 9](#), the execution time of all methods is reported. Based on these values, it can be seen that the proposed method is not the fastest. This issue is quite logical, considering the iterative structure of the ant algorithm. Because in this algorithm, we have considered 30 iterations to get the best feature. Also, in each iteration, reinforcement learning performs, which causes increasing the execution time.

We have also calculated the computation complexity of the proposed algorithm. [Table 10](#) shows the computational complexity of different steps of the proposed method, considering the pseudo-code presented in [Algorithm 2](#).

## 6. Conclusion

To the best of our knowledge, no ACO-based semi-supervised feature selection technique has been proposed to date. The proposed algorithm, SemiACO, utilizes nonlinear reinforcement learning to learn ACO heuristic functions. The proposed algorithm computes relevancy between features and the class label of labeled samples (supervised) and redundancy between all features (unsupervised). SemiACO uses unlabeled and unlabeled sample information simultaneously to find the most profitable features. The classifier outcomes illustrate that the recommended technique is better than other strategies on 14 datasets. The results show that

the proposed method has high accuracy with reasonable execution time, in two cases of random labeling (20 % and 40 % of the training samples). Given the excellent performance of SemiACO, we intend to generalize this approach to other semi-supervised feature selection problems such as multi-label semi-supervised feature selection. Also, other representative computational intelligence algorithms can be used to solve the feature selection problems, like monarch butterfly optimization (MBO) (G. G. Wang, Deb, & Cui, 2019), earthworm optimization algorithm (EWA) (Wang, Deb, & dos Santos Coelho, 2018), elephant herding optimization (EHO) (Li, Chen, Wang, Heidari, & Mirjalili, 2020), moth search (MS) algorithm (Wang, 2018), Slime mould algorithm (SMA) (Li et al., 2020), hunger games search (HGS) (Yang, Chen, Heidari, & Gandomi, 2021), colony predation algorithm (CPA) (Tu, Chen, Wang, & Gandomi, 2021), and Harris hawks optimization (HHO) (Heidari et al., 2019). We intend to examine these algorithms in the future.

## CRediT authorship contribution statement

**Fereshteh Karimi:** Software, Writing – original draft. **Mohammad Bagher Dowlatshahi:** Conceptualization, Data curation, Investigation, Methodology, Supervision, Writing – review & editing. **Amin Hashemi:** Data curation, Methodology, Supervision, Validation, Writing – review & 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

Data will be made available on request.

## References

- Baird, G. (2020). Optimising darts strategy using Markov decision processes and reinforcement learning. *Journal of the Operational Research Society*, 71(6), 1020–1037. <https://doi.org/10.1080/01605682.2019.1610341>
- Bäuerle, N., & Rieder, U. (2010). Markov Decision Processes. *Jahresbericht Der Deutschen Mathematiker-Vereinigung*, 112(4), 217–243. <https://doi.org/10.1365/s13291-010-0007-2>
- Bayati, H., Dowlatshahi, M. B., & Hashemi, A. (2022). MSSL: A memetic-based sparse subspace learning algorithm for multi-label classification. *International Journal of Machine Learning and Cybernetics*. <https://doi.org/10.1007/s13042-022-01616-5>
- Bayati, H., Dowlatshahi, M. B., & Paniri, M. (2020). MLPSO: A Filter Multi-label Feature Selection Based on Particle Swarm Optimization. In *2020 25Th International Computer Conference, Computer Society of Iran (CSICC)* (pp. 1–6). <https://doi.org/10.1109/CSICC49403.2020.9050087>
- Beiranvand, F., Mehrdad, V., & Dowlatshahi, M. B. (2022). Unsupervised feature selection for image classification: A bipartite matching-based principal component analysis approach. *Knowledge-Based Systems*, 250, Article 109085. <https://doi.org/10.1016/j.knosys.2022.109085>
- Benabdeslem, K., & Hindawi, M. (2011). Constrained Laplacian score for semi-supervised feature selection. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [https://doi.org/10.1007/978-3-642-23780-5\\_23](https://doi.org/10.1007/978-3-642-23780-5_23)
- Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems With Applications*, 42(22), 8520–8532. <https://doi.org/10.1016/j.eswa.2015.07.007>
- Cao, D., Chen, Y., Chen, J., Zhang, H., & Yuan, Z. (2021). An improved algorithm for the maximal information coefficient and its application. *Royal Society Open Science*, 8(2). <https://doi.org/10.1098/rsos.201424>
- Chen, K., Xue, B., Zhang, M., & Zhou, F. (2022). Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 26(3). <https://doi.org/10.1109/TEVC.2021.3100056>
- Cheng, F., Chu, F., Xu, Y., & Zhang, L. (2021). A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection. *IEEE Transactions on Cybernetics*. <https://doi.org/10.1109/TCYB.2021.3053944>
- Dalvand, A., Dowlatshahi, M. B., & Hashemi, A. (2022). SGFS: A semi-supervised graph-based feature selection algorithm based on the PageRank algorithm. In *2022 27Th International Computer Conference, Computer Society of Iran (CSICC)* (pp. 1–6). <https://doi.org/10.1109/CSICC55295.2022.9780486>

- Deng, X., Li, M., Wang, L., & Wan, Q. (2021). *RFCBF: enhance the performance and stability of Fast Correlation-Based Filter*. <https://doi.org/10.48550/arxiv.2105.14519>.
- Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. *Multimedia Tools and Applications*, 78(3), 3797–3816. <https://doi.org/10.1007/s11042-018-6083-5>
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2–3), 243–278. <https://doi.org/10.1016/j.tcs.2005.05.020>
- Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66. <https://doi.org/10.1109/4235.585892>
- Dowlatshahi, M. B., Kuchaki Rafsanjani, M., & Gupta, B. B. (2021). An energy aware grouping memetic algorithm to schedule the sensing activity in WSNs-based IoT for smart cities. *Applied Soft Computing*, 108, Article 107473. <https://doi.org/10.1016/j.asoc.2021.107473>
- Dowlatshahi, M. B., & Nezamabadi-pour, H. (2014). GGSAs: A grouping gravitational search algorithm for data clustering. *Engineering Applications of Artificial Intelligence*, 36, 114–121. <https://doi.org/10.1016/j.engappai.2014.07.016>
- Dowlatshahi, M. B., Nezamabadi-Pour, H., & Mashinchi, M. (2014). A discrete gravitational search algorithm for solving combinatorial optimization problems. *Information Sciences*, 258, 94–107. <https://doi.org/10.1016/j.ins.2013.09.034>
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1), 86–92.
- Hashemi, A., & Dowlatshahi, M. B. (2022). An Ensemble Of Feature Selection Algorithms Using OWA Operator. In *2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)* (pp. 1–6). <https://doi.org/10.1109/CFIS54774.2022.9756456>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality. *Expert Systems with Applications*, 142, Article 113024. <https://doi.org/10.1016/j.eswa.2019.113024>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MFS-MCDM: Multi-label feature selection using multi-criteria decision making. *Knowledge-Based Systems*, 206, Article 106365. <https://doi.org/10.1016/j.knosys.2020.106365>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021a). Gravitational Search Algorithm. In *Handbook of AI-based Metaheuristics* (p. 32).
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2021b). Minimum redundancy maximum relevance ensemble feature selection: A bi-objective Pareto-based approach. *Journal of Soft Computing and Information Technology*.
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2021c). A bipartite matching-based feature selection for multi-label learning. *International Journal of Machine Learning and Cybernetics*, 12(2), 459–475. <https://doi.org/10.1007/s13042-020-01180-w>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021d). VMFS: A VIKOR-based multi-target feature selection. *Expert Systems with Applications*, 182, Article 115224. <https://doi.org/10.1016/j.eswa.2021.115224>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021e). An efficient Pareto-based feature selection algorithm for multi-label classification. *Information Sciences*, 581, 428–447. <https://doi.org/10.1016/j.ins.2021.09.052>
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2022). Ensemble of feature selection algorithms: a multi-criteria decision-making approach. *International Journal of Machine Learning and Cybernetics*, 13(1), 49–69. <https://doi.org/10.1007/S13042-021-01347-Z>
- Hashemi, A., Joodaki, M., Joodaki, N. Z., & Dowlatshahi, M. B. (2022). Ant Colony Optimization equipped with an ensemble of heuristics through Multi-Criteria Decision Making: A case study in ensemble feature selection. *Applied Soft Computing*, 10946. <https://doi.org/10.1016/j.asoc.2022.109046>
- Hashemi, A., Pajoohan, M.-R., & Dowlatshahi, M. B. (2022). Online streaming feature selection based on Sugeno fuzzy integral. *2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 1–6. <https://doi.org/10.1109/CFIS54774.2022.9756477>
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97. <https://doi.org/10.1016/j.future.2019.02.028>
- Hindawi, M., Elghazel, H., & Benabdeslem, K. (2013). *Efficient semi-supervised feature selection by an ensemble approach* (pp. 41–55). In Proc. Int. Workshop Complex Mach. Learn. Problems Ensemble Methods.
- JAFFE Dataset | Papers With Code. (n.d.). Retrieved March 6, 2022, from <https://paperswithcode.com/dataset/jaffe>.
- Jia, Y., Kwong, S., Hou, J., & Wu, W. (2020). Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7), 2510–2521. <https://doi.org/10.1109/TNNLS.2019.2933223>
- Joodaki, M., Dowlatshahi, M. B., & Joodaki, N. Z. (2021). An ensemble feature selection algorithm based on PageRank centrality and fuzzy logic. *Knowledge-Based Systems*, 233, Article 107538. <https://doi.org/10.1016/j.knosys.2021.107538>
- Joshi, D. J., Kale, I., Gandewar, S., Korate, O., Patwari, D., & Patil, S. (2021). Reinforcement Learning: A Survey. *Advances in Intelligent Systems and Computing*, 1311 AISC, 297–308. [https://doi.org/10.1007/978-981-33-4859-2\\_29](https://doi.org/10.1007/978-981-33-4859-2_29)
- Reinforcement learning: A survey, 4 Journal of Artificial Intelligence Research 237 (1996). <https://doi.org/10.1613/jair.301>.
- Kanan, H. R., Faez, K., & Taheri, S. M. (2007). Feature selection using Ant Colony Optimization (ACO): A new method and comparative study in the application of face recognition system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4597 LNCS, 63–76. [https://doi.org/10.1007/978-3-540-73435-2\\_6](https://doi.org/10.1007/978-3-540-73435-2_6).
- Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., ... Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6). <https://doi.org/10.1038/89044>
- Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11), 1238–1274. <https://doi.org/10.1177/0278364913495721>
- Lee Rodgers, J., Wander, A. N., & W.. (1988). Thirteen ways to look at the correlation coefficient. *American Statistician*, 42(1), 59–66. <https://doi.org/10.1080/00031305.1988.10475524>
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, 50(6). <https://doi.org/10.1145/3136625>
- Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111. <https://doi.org/10.1016/j.future.2020.03.055>
- Liao, Y., & Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers and Security*, 21(5), 439–448. [https://doi.org/10.1016/S0167-4048\(02\)00514-X](https://doi.org/10.1016/S0167-4048(02)00514-X)
- Liu, Y., Nie, F., Wu, J., & Chen, L. (2010). Semi-supervised feature selection based on label propagation and subset selection. *Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application*, 293–296. <https://doi.org/10.1109/ICCIA.2010.6141595>
- Ma, Z., Yang, Y., Nie, F., Uijlings, J., & Sebe, N. (2011). Exploiting the entire feature space with sparsity for automatic image annotation. *MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops*, 283–292. <https://doi.org/10.1145/2072298.2072336>
- Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers and Operations Research*, 134. <https://doi.org/10.1016/j.cor.2021.105400>
- Miao, J., & Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, 91, 919–926. <https://doi.org/10.1016/j.procs.2016.07.111>
- Miri, M., Dowlatshahi, M. B., & Hashemi, A. (2022a). Evaluation multi label feature selection for text classification using weighted borda count approach. *2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 1–6. <https://doi.org/10.1109/CFIS54774.2022.9756467>
- Miri, M., Dowlatshahi, M. B., & Hashemi, A. (2022b). Feature selection for multi-label text data: An ensemble approach using geometric mean aggregation. *2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 1–6. <https://doi.org/10.1109/CFIS54774.2022.9756484>
- Miri, M., Dowlatshahi, M. B., Hashemi, A., Rafsanjani, M. K., Gupta, B. B., & Alhalabi, W. (2022). Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*. <https://doi.org/10.1002/int.23044>
- Monteiro, M. S. R., Fontes, D. B. M. M., & Fontes, F. A. C. C. (2012). Ant Colony Optimization: a literature survey. *FEPE Working Papers*, December, undefined-undefined.
- NCI60: NCI 60 Data in ISLR: Data for an Introduction to Statistical Learning with Applications in R. (n.d.). Retrieved March 11, 2022, from <https://rdr.io/cran/ISLR/man/NCI60.html#heading-0>.
- Pan, W. (2021). Feature Selection Algorithm Based on Maximum Information Coefficient. *IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2600–2603. <https://doi.org/10.1109/IAEAC50856.2021.9390868>
- Panir, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowledge-Based Systems*, 192, Article 105285. <https://doi.org/10.1016/j.knosys.2019.105285>
- Panir, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021). Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection. *Swarm and Evolutionary Computation*, 64, Article 100892. <https://doi.org/10.1016/j.swevo.2021.100892>
- Roughened Random Forests for Binary Classification - ProQuest. (n.d.). Retrieved March 7, 2022, from <https://www.proquest.com/openview/11be8247a8128deb0e26eb2d1355708e/1?pq-origsite=gscholar&cbl=18750>.
- Sechidis, K., & Brown, G. (2018). Simple strategies for semi-supervised feature selection. *Machine Learning*, 107(2), 357–395. <https://doi.org/10.1007/s10994-017-5648-2>
- Sheikhpour, R., Sarram, M. A., Gharaghani, S., & Chahooki, M. A. Z. (2017). A Survey on semi-supervised feature selection methods. *Pattern Recognition*, 64, 141–158. <https://doi.org/10.1016/j.patcog.2016.11.003>
- Sheikhpour, R., Sarram, M. A., & Sheikhpour, E. (2018). Semi-supervised sparse feature selection via graph Laplacian based scatter matrix for regression problems. *Information Sciences*, 468, 14–28. <https://doi.org/10.1016/j.ins.2018.08.035>
- Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-019-09682-y>
- Song, X. F., Zhang, Y., Gong, D. W., & Gao, X. Z. (2021). A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Transactions on Cybernetics*. <https://doi.org/10.1109/TCYB.2021.3061152>
- Song, X. F., Zhang, Y., Guo, Y. N., Sun, X. Y., & Wang, Y. L. (2020). Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Transactions on Evolutionary Computation*, 24(5). <https://doi.org/10.1109/TEVC.2020.2968743>
- Sugiyama, M., Idé, T., Nakajima, S., & Sese, J. (2010). Semi-supervised local Fisher discriminant analysis for dimensionality reduction. *Machine Learning*, 78(1–2), 35–61. <https://doi.org/10.1007/s10994-009-5125-7>

- Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1), 181–211. [https://doi.org/10.1016/S0004-3702\(99\)00052-1](https://doi.org/10.1016/S0004-3702(99)00052-1)
- Tu, J., Chen, H., Wang, M., & Gandomi, A. H. (2021). The colony predation algorithm. *Journal of Bionic Engineering*, 18(3). <https://doi.org/10.1007/s42235-021-0050-y>
- UCI Machine Learning Repository: Musk (Version 2) Data Set. (n.d.). Retrieved March 7, 2022, from <https://archive.ics.uci.edu/ml/datasets/Musk%2B%28Version%2B2%29>
- van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440.
- van Seijen, H., Mahmood, A. R., Pilarski, P. M., Machado, M. C., & Sutton, R. S. (2016). True online temporal-difference learning. *Journal of Machine Learning Research*, 17, 1–40.
- Venkatesh, B., & Anuradha, J. (2019). A review of Feature Selection and its methods. *Cybernetics and Information Technologies*, 19(1), 3–26. <https://doi.org/10.2478/CAIT-2019-0001>
- Wang, G. G. (2018). Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 10(2). <https://doi.org/10.1007/s12293-016-0212-3>
- Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural Computing and Applications*, 31(7). <https://doi.org/10.1007/s00521-015-1923-y>
- Wang, G. G., Deb, S., & dos Santos Coelho, L. (2018). Earthworm optimisation algorithm: A bio-inspired metaheuristic algorithm for global optimisation problems.
- International Journal of Bio-Inspired Computation, 12(1). <https://doi.org/10.1504/ijbic.2018.093328>
- Wang, Y., Wang, J., Liao, H., & Chen, H. (2017). An efficient semi-supervised representatives feature selection algorithm based on information theory. *Pattern Recognition*, 61, 511–523. <https://doi.org/10.1016/j.patcog.2016.08.011>
- Yang, Y., Chen, H., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, 177. <https://doi.org/10.1016/j.eswa.2021.114864>
- Zhao, F., He, X., & Wang, L. (2021). A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem. *IEEE Transactions on Cybernetics*, 51(11). <https://doi.org/10.1109/TCYB.2020.3025662>
- Zhao, F., Zhang, L., Cao, J., & Tang, J. (2021). A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem. *Computers and Industrial Engineering*, 153. <https://doi.org/10.1016/j.cie.2020.107082>
- Zhou, S., Xing, L., Zheng, X., Du, N., Wang, L., & Zhang, Q. (2021). A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times. *IEEE Transactions on Cybernetics*, 51(3). <https://doi.org/10.1109/TCYB.2019.2939219>
- Zhu, H., You, X., & Liu, S. (2019). Multiple ant colony optimization based on pearson correlation coefficient. *IEEE Access*, 7, 61628–61638. <https://doi.org/10.1109/ACCESS.2019.2915673>