



# Socio-inspired evolutionary algorithms: a unified framework and survey

Laxmikant Sharma<sup>1</sup> · Vasantha Lakshmi Chellapilla<sup>1</sup> · Patvardhan Chellapilla<sup>2</sup>

Accepted: 11 February 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

It is well recognized that human societies and their problem-solving capabilities have evolved much faster than biological evolution. The inspiration from human behaviors, knowledge exchange, and transformation has given rise to a new evolutionary computation paradigm. Multiple research endeavors have been reported in the literature inspired by the diverse aspects of human societies with corresponding terminologies to describe the algorithm. These endeavors have resulted in piles of algorithms, worded differently but with more or less similar underlying mechanisms causing immense confusion for a new reader. This paper presents a generalized framework for these socio-inspired evolutionary algorithms (SIEAs) or socio-inspired meta-heuristic algorithms. A survey of various SIEAs is provided to highlight the working of these algorithms on a common framework, their variations and improved versions proposed in the literature, and their applications in various fields of search and optimization. The algorithmic description and the comparison of each SIEA with the general framework enable a clearer understanding of the similarities and differences between these methodologies. Efforts have been made to provide an extensive list of references with due context. This paper could become an excellent reference as a starting point for anyone interested in this fascinating field of research.

**Keywords** Evolutionary computation · Socio-inspired evolutionary algorithms · Socio-inspired meta-heuristic algorithms · Soft computing

## 1 Introduction

Optimization is an essential task in engineering and many other aspects of life and living. Various optimization techniques have been proposed in the literature to solve such problems. These techniques are classified into two broad categories—exact and heuristics. Exact or deterministic methods comprise the classical approaches founded on mathematical and dynamic programming models and can solve

optimization problems exactly. However, they may require infeasible computational time with increased problem size and complexity. Since real-world optimization problems are typically nonlinear, multi-modal, vast, dynamic, and complex, exact algorithms are not practically suitable to solve them. Heuristics and meta-heuristics are usually employed in such cases because of their user-friendly implementation nature and schematic algorithmic framework and design. Heuristics are designed for a specific problem, whereas meta-heuristics have broad applicability.

✉ Laxmikant Sharma  
lkrdkrishnan@gmail.com

Vasantha Lakshmi Chellapilla  
vasanthalakshmi@dei.ac.in

Patvardhan Chellapilla  
cpatvardhan@dei.ac.in

<sup>1</sup> Department of Physics and Computer Science, Dayalbagh Educational Institute, Dayalbagh, Agra 282005, Uttar Pradesh, India

<sup>2</sup> Department of Electrical Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra 282005, Uttar Pradesh, India

### 1.1 Evolutionary algorithms

Several meta-heuristics are nature-inspired algorithms. Evolutionary algorithms (EAs) are one of the leading meta-heuristics. EAs drew inspiration from observing flora and fauna closely to understand their relationship with their environment. Darwin (1859, 2018) explained this relationship in 1859 and called it natural selection. According to his theory, individuals not well adapted to their environment do not survive long enough to reproduce or have fewer

chances to reproduce than individuals of the same species that have acquired beneficial characteristics through variation during their production. This adaptation of the individuals to their environment is called their fitness; an individual with higher fitness is more adapted to the environment. Each algorithm in the EA class is a population-based, guided random meta-heuristic algorithm. These algorithms simulate species' natural-biological evolution concepts such as reproduction, mutation, recombination, selection, migration, locality, and neighborhood. Evolutionary programming (EP) (Fogel 1997), evolution strategies (ES) (Rechenberg 1970), and genetic algorithms (GA) (Goldberg 1989) are some notable EA techniques.

Akin to EAs, there are other classes of meta-heuristics. Swarm intelligence (SI) algorithms were designed, taking inspiration from the collective intelligence of insects, birds, and fish. SI works on social behavior and the self-organizing nature of various natural or artificial agents. Particle swarm optimization (PSO) (Kennedy and Eberhart 1995), ant colony optimization (ACO) (Dorigo and Di Caro 1999), and artificial bee colony (ABC) optimization (Karaboga and Basturk 2007) are some notable swarm-based algorithms. Physics- and chemistry-based algorithms are inspired by phenomena seen in physical and chemical Sciences, respectively (Biswas et al. 2013; Siddique and Adeli 2017). Some good survey articles on bio-inspired meta-heuristics are Darwish (2018) and Fan et al. (2020).

Memetic algorithms (MAs) introduced local search techniques at specific parts of evolutionary algorithms to improve their performance (Moscato 1989; Norman et al. 1991). These algorithms are inspired by complex natural systems where a population of individuals evolves through evolutionary processes and individual learning. This learning or imitation of behavior and knowledge from other individuals is represented as memes—a cultural equivalent of genes, introduced by Trivers and Dawkins (1976). Memetic algorithms utilize Lamarckian, and Baldwinian learning (Whitley et al. 1994) and are also known as Lamarckian EAs and Baldwinian EAs. MAs have been successfully applied to solve real-world problems and have shown high performance (Neri and Cotta 2012).

Reynolds (1994) proved that introducing a belief space in a general GA framework makes it more powerful. He developed a framework of cultural algorithms in which experiences of the candidate solutions are exchanged with the help of belief space. This cultural framework has been improved and hybridized many times. Some conspicuous algorithms inspired by this framework are cultural algorithm with evolutionary programming (Chung and Reynolds 1998), cultural swarms (Iacoban et al. 2003), cultured differential evolution (Becerra and Coello 2006), social learning optimization (Liu et al. 2016) and others.

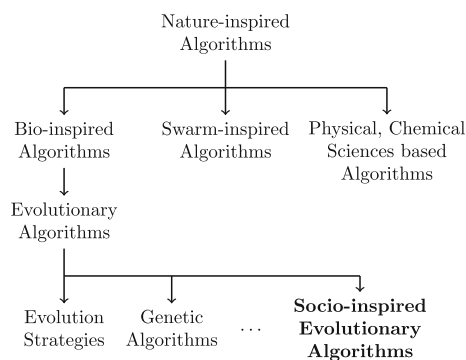


Fig. 1 Taxonomy of SIEAs

## 1.2 Socio-inspired evolutionary algorithms

It is well accepted that human societies and humans' problem-solving capabilities have evolved much faster than biological evolution (Bjorklund et al. 2010). This evolution is due to humans' biological structure and social means of living and surviving. DNA studies reveal that a newborn child is not genetically much different from his ancestors that were born 40,000 years ago (Varki and Altheide 2005), i.e., human's hardware part is substantially the same. However, socially, humans have left the caves and are planning to emerge from the surface of Earth to colonize other planets in the next few decades, i.e., the software part has immensely changed over time. With the evolution of human life on this planet, humans started living in groups or social organizations for convenience, resource sharing, collective problem-solving, and division of workload and responsibilities. Social organizations of humans can be regarded eventually as an optimization process for the overall welfare of individuals and the organization. Individuals in societies learn to live and survive from their parents and other older members. This behavior and knowledge transfer among members is pivotal for the evolution of societies. Occasionally, this influence works in reverse order, i.e., older members can also learn from their descendants.

The social phenomena of human interaction, behavioral exchange, learning mechanisms, and others have motivated a new class of meta-heuristics dubbed socio-inspired evolutionary algorithms (SIEAs) or socio-inspired meta-heuristics (see Fig. 1).

Many research endeavors have been reported in the literature claiming inspiration from different social phenomena. These endeavors have resulted in a plethora of algorithms worded differently from each other. However, the underlying mechanisms could be more or less similar, causing immense confusion to a new reader. One of the first serious discussions and analyses of SIEAs emerged during the 2000s with the work of Neme and Hernández (2009). They classified these algorithms into four categories: leadership,

alliance formation, social labeling, and neighborhood delimitation and segregation. Similarly, Khuat et al. (2016), and Kumar and Kulkarni (2019) surveyed many SIEAs. Khuat et al. (2016) provided a limited survey without describing several powerful SIEAs. Kumar and Kulkarni (2019) provided detailed descriptions of each algorithm independently. However, these previously published studies are limited to individual surveys without any effort to put reported SIEAs into a common framework. The algorithms surveyed in these articles are selected based on the social jargon used. Moreover, only some algorithms qualify the necessary criteria of the social grouping for being an SIEA. However, only social tagging is utilized to discriminate the types of individuals in them.

This paper proposes a common framework for describing the prominent SIEAs. In addition, a survey is provided on the state-of-the-art of SIEAs. For each SIEA, the pseudocode description indicates points of similarity and distinctiveness in the context of the broad framework provided. A detailed analysis of each algorithm highlights how the basic steps of EAs and SIEAs, viz. population initialization, selection, recombination, exploration, and exploitation, are incorporated into these algorithms. This description cuts through the diverse terminology of each algorithm to present it in an easily comprehensible manner. A comparison table that clarifies the similarities and differences between surveyed SIEAs is also furnished.

Furthermore, a detailed table lists the applications to which each algorithm has been applied. An extensive list of references has been provided to cover the significant publications in the area. Therefore, this paper could become an excellent starting point for a researcher trying to understand the field.

The rest of the paper is organized as follows. The following section—Sect. 2—presents an introduction to social phenomena and the division of SIEAs according to well-known social phenomena classification. In Sect. 3, the unified framework on SIEAs is proposed and discussed. Then prior art on SIEAs is given in section 4. Section 5 lists the application areas of these algorithms. Finally, Sect. 6 concludes the paper and provides future directions for researchers.

## 2 Human social phenomena

Human society provides a complex system with diverse ideas of optimization working in parallel and serial. These societies are built on the conventions of living in peace and working to fulfill everyone's needs for food, clothes, and home. The basic needs of a human being are significantly achieved with various social activities or social actions. Interactions in humans are primal social activities that lead to the transfer of information. This information is usually utilized to general-

**Table 1** Classification of Social Phenomena (Giddings 1914)

Category	Social Phenomena
Cultural	Lingual, animistic aesthetic, animistic religious, scientific ideas, ceremonial use of manners, dress and festivities, music and dancing, games, poetic and plastic arts, religious exercises, exploration and research, and others
Economic	Economic cooperation and competition, financial excitements, agriculture, mining, fisheries, manufacturing, commerce, transportation, finance, and others
Moral and Juristic	Revenge, property, and marriage rights, moral and juristic traditions and customs, oaths, pledges, fines and compensations, boycotting, hazing, mobbing, lynching, and others
Political	Self-existence of the group, citizenship and kinship, power, splendor, pleasure, prosperity, righteousness, liberty, equality, enlightenment, coercion, bribery, patronage, loyalty, and others

ize facts and to benefit society. Although social interactions are not based on any overarching optimizing principle, they lead to robust societal structures.

To improve their lifestyle, humans observe other similar or distinct species and change their behavior and actions accordingly. These observations also influence people to do numerous activities of interest. Furthermore, humans self-observe their actions and try to improve in the future following the previous results. These social means of behavior and other qualities, including information exchange and self-observation, are examples of social phenomena. The algorithms inspired by ideas of social phenomena have been shown to perform well in practice for various optimization problems. Compared to other swarm intelligence mechanisms, algorithms inspired by human society are more robust (Neme and Hernández 2009). Other common social phenomena include collective behavior, self-organizing systems, social interaction, coordination, cooperation and competition, information propagation, communication, language, relationships, leadership, politics, elections, sports, tourism, disease spread, consensus, and social opinion, to name a few. These social phenomena are categorized by various sociologists (Hayes 1911). One of the most influential classifications is provided by Giddings (1914) into cultural, economic, moral and juristic, and political categories. Table 1 lists common phenomena in each category.

This paper utilizes the classification proposed by Giddings to organize SIEAs into various categories for easy comprehension. Table 2 provides a list of SIEAs, their inspirational ideas, and related social phenomena. From the table, it can

**Table 2** State-of-art SIEAs, their inspirational idea, and corresponding social phenomena

SIEA	Inspired by	Social phenomena
<i>Cultural</i>		
Society and civilization algorithm (Ray and Liew 2003)	Social behavior of human societies to improve themselves	Social interaction & cooperation
Soccer league competition algorithm (Moosavian and Roodsari 2013)	Competition within and among soccer teams	Social competition; Sports
Socio evolution & learning optimization (Kumar et al. 2018)	Social learning behavior of family individuals from other members and families	Social learning
Nomadic people optimizer (Salih and Alsewari 2020)	Behavior of nomads in the search for food and water	Search for livable environment
<i>Economic</i>		
Imperialist competitive algorithm (Atashpaz-Gargari and Lucas 2007b)	Imperialistic competition among empires to improve themselves	Imperialistic competition
<i>Moral and Juristic</i>		
–		
<i>Political</i>		
Parliamentary optimization (Borji 2007; Borji and Hamidi 2009)	Behavior of political party members in a parliament head election process	Political competition
Election campaign optimization (Lv et al. 2010)	Influence of candidates on voters in election campaign	Political influence
Group leaders optimization (Daskin and Kais 2011b)	Influence of the leaders and better members in social groups	Self-existence of the group; Political influence
Election algorithm (Emami and Derakhshan 2015)	Behavior of candidates in presidential elections	Politics
Ideology algorithm (Huan et al. 2017)	Competitive behavior of party members to improve their ranking in the party	Social competition
Political optimizer (Askari et al. 2020)	Multiple phases of politics	Politics

be seen that no SIEA is proposed based on moral and juristic social phenomena.

### 3 Framework for SIEAs

Exploration and exploitation are two fundamental operations in optimization meta-heuristics. In the search for better solutions, the feasible solution space is explored in the exploration process. However, in exploitation, solution space is searched around a good solution under consideration to find more promising solutions in its vicinity. Diversification is also used for exploration, whereas intensification is also used for exploitation. A proper balance in the computational effort devoted to these two operations is critical to the success of any meta-heuristic (Bäck et al. 1997). The selection, crossover, and mutation-like operators typically perform these operations. It is not within this paper's scope to discuss the correspondence between the operator and its operation. Please refer Eiben and Schippers (1998); Črepinšek et al. (2013); Cuevas et al. (2021) for details.

The essential idea in SIEAs is to divide the population into subpopulations, each subpopulation searching in a differ-

ent search sub-space while simultaneously maintaining the number of individuals in subpopulations commensurate with the importance of the search sub-space. Initially, individuals in the search space are divided according to a predefined method. Gradually, these individuals migrate to the promising regions in the search space. Thus, well-designed SIEAs typically start with a higher exploration rate and progressively increase exploitation in later runs.

In terms of individual representation, fitness evaluation, selection mechanisms, and initial population diversity, SIEAs are analogous to conventional EAs. However, operators in SIEAs simulate social phenomena rather than natural-biological evolution. In each run of SIEAs, an increasing number of solution individuals are allocated to the most promising sub-regions of the solution space. These algorithms divide the feasible search space into subclasses/clusters/groups based on societal organizations like families, villages, colonies, states, and others, as seen in the real world. The non-duplication of the population individuals in these groups is not guaranteed, and these groups can overlap. The best-fitted individuals in every group are appointed as local leaders, and the best individual among all the groups becomes the global leader in each iteration. An individual

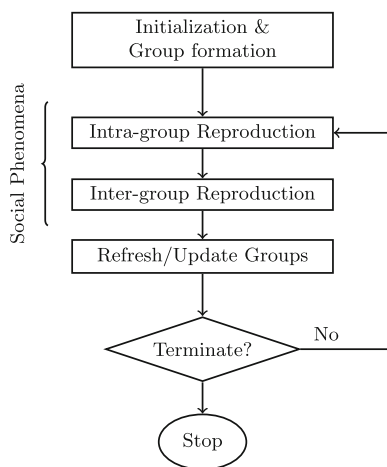


Fig. 2 Flowchart of a generic SIEA

improves himself by interacting and simulating the characteristics of other better individuals. Specifically, an individual gains influence from peer members (in the same group and sometimes from another group), local or group leaders, and global leaders. A group with the best individuals is the best group and usually provides the optimal global solution. A flowchart and a pseudocode description of a generic SIEA are succinctly provided in Fig. 2 and Algorithm 1.

**Algorithm 1** Pseudocode of a generic SIEA

- 1: Initialize parameters and randomly generate candidate solutions
- 2: Create logical groups of solutions
- 3: **while** termination condition not satisfied **do**
- 4:   Intra-group reproduction
- 5:   Inter-group reproduction
- 6:   Update groups
- 7: **end while**

**3.1 Algorithmic components**

The two main components of SIEAs are social grouping mechanisms and social phenomena-inspired operators. The initialization in SIEAs depends upon the social grouping mechanism used in it. Best individual (leader) selection, evaluation of individuals, number of individuals in a group, and group updating mechanisms are other components that depend on the choice of the social grouping mechanism. These and other algorithmic ingredients that define a new SIEA in line with the above pseudocode are the following.

1. *Initialization*: A population of  $N$  individuals is typically generated randomly to initialize an SIEA. This initialization of population individuals can also be seeded using some heuristics for a better start. Then, the initial population is divided into numerous groups of individuals.

The best individual in each group is declared the group leader, and the best individual in all groups is declared the global leader. This global leader represents the best-found solution in an SIEA.

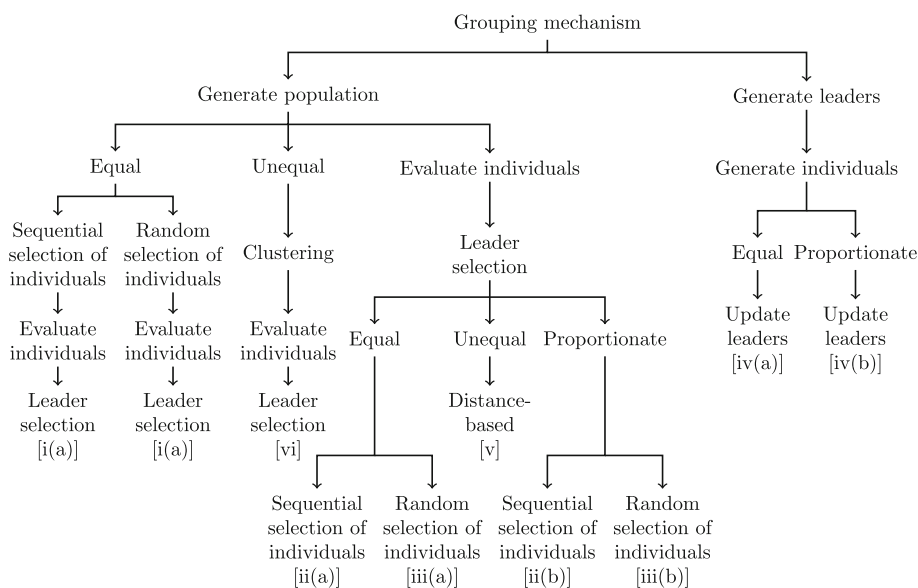
A. *Social Organization or Grouping*: The population in SIEAs is typically divided into  $M$  mutually exclusive but collectively exhaustive subpopulations or groups. An individual can also be assigned to more than one group. Individuals may be selected randomly, sequentially, or by a distance-based clustering technique for this division. Senadji and Dawes (2010) argued that the groups generated using clustering methods are more prone to social loafing than the randomly allocated groups. The generated subpopulations can possess an equal or varying number of individuals depending upon the selection method employed. Group leaders are selected based on their fitness evaluation as the best individuals in the group. Several grouping mechanisms utilized in the SIEAs are as follows.

- i. Generate  $m \times l = n$  individuals randomly, i.e.,  $N = \{I_1, I_2, \dots, I_n\}$ , where  $m$  is the total number of groups,  $l$  is the number of individuals in each group and  $I_i$  represents the  $i^{\text{th}}$  individual.
  - (a) Create subpopulations by grouping first  $l$  individuals, i.e.,  $\{I_1, I_2, \dots, I_l\}$  in the first group, next  $\{I_{l+1}, I_{l+2}, \dots, I_{2l}\}$  individuals in the second group, ..., and last  $\{I_{(m-1)l+1}, I_{(m-1)l+2}, \dots, I_{ml}\}$  individuals in the  $m^{\text{th}}$  group. The whole population can also be sorted according to their fitness before grouping to generate groups with phenotypically similar individuals.
  - (b) Create subpopulations by selecting  $l$  individuals randomly for the first group, selecting  $l$  individuals randomly from the remaining population for the next group, and so on.
- ii. Generate  $n$  individuals randomly, i.e.,  $N = \{I_1, I_2, \dots, I_n\}$ . Evaluate each population individual  $I_i$  according to the objective function. Select the best  $m$  individuals to be the leaders of respective  $m$  groups  $\{M_1, M_2, \dots, M_m\}$ . Sequentially assign the rest  $n - m$  individuals to respective groups/leaders -
  - (a) Equally, i.e., assign first  $\frac{n-m}{m} = l$  individuals in the first group, second  $l$  individuals in the next group, and so on.
  - (b) In proportion to fitness of the corresponding, i.e., the number of individuals in a group  $M_i$  equals  $\text{round} \left\{ \left| \frac{f(L_i)}{\sum f(L_i)} \right| \cdot (n - m) \right\}$ , where  $f(L_i)$  represents  $i^{\text{th}}$  leader's fitness. Hence, a better leader will lead more individuals.

**Table 3** A comparison of social grouping mechanisms used in SIEAs

#	Grouping Mechanism			Intensification	Diversification
	Leader selection	Individual selection	Member count		
i(a)	After grouping	Sequential	Equal	Low	High
i(b)	After grouping	Random	Equal	Low	High
ii(a)	Prior grouping	Sequential	Equal	Low	Moderate
ii(b)	Prior grouping	Sequential	Unequal, proportionate	Low	Moderate
iii(a)	Prior grouping	Random	Equal	Low	High
iii(b)	Prior grouping	Random	Unequal, proportionate	Low	High
iv(a)	Generate leaders	Generate individuals	Equal	High	Low
iv(b)	Generate leaders	Generate individuals	Unequal, proportionate	High	Low
v	Prior grouping	sequential/random	Unequal, distance-based	High	Low
vi	After grouping (clustering)	sequential/random	Unequal, distance-based	High	Low

**Fig. 3** Schematic of the grouping mechanisms



- This mechanism is called fitness-proportionate group formation.
- iii. Generate  $n$  individuals randomly. Evaluate each population individual  $I_i$  according to the objective function. Select the best  $m$  individuals to be the leaders of respective  $m$  groups  $\{M_1, M_2, \dots, M_m\}$ . Randomly assign the rest  $n - m$  individuals
    - (a) Equally, i.e., select  $\frac{n-m}{m} = l$  individuals randomly from the rest  $n - m$  individuals and assign them to some group  $M_i$ , again select  $l$  individuals from the remaining  $n - m - l$  individuals and assign them to another group, and so on. This mechanism leads to equal-sized groups with randomly selected individuals.

- (b) In proportion to fitness of the corresponding leader, i.e., the number of individuals in a group  $M_i$  equals  $round \left\{ \left\lfloor \frac{f(L_i)}{\sum f(L_i)} \cdot (n - m) \right\rfloor \right\}$ .
- iv. Generate  $m$  individuals randomly and designate them as leaders. Populate other members in each subpopulation by mutating the leaders appropriately.
  - (a) Populate groups with an equal number of individuals.
  - (b) Populate groups with the individuals in the proportion of its leaders' fitness.
 Each group's leaders are re-selected at the end of this grouping mechanism.
- v. Generate  $n$  individuals randomly. Evaluate each population individual  $I_i$  according to the objective function. Select the best  $m$  individuals to be the leaders of respective  $m$  groups  $\{M_1, M_2, \dots, M_m\}$ . Sequentially or randomly select an indi-

vidual  $I_i \in \{I_1, I_2, \dots, I_{n-m}\}$  and assign it to some group  $M_i$  whose leader  $L_i$  is more similar to the selected individual  $I_i$ , i.e.,  $L_i$  is the nearest neighbor to  $I_i$ .

- vi. Generate  $n$  individuals randomly. Select an individual  $I_i$  sequentially or randomly from this population and assign it to a cluster  $C_g$ , i.e.,  $C_g \leftarrow I_i$ ;  $g = 1, \dots, k$  from which it has minimum Euclidean distance than from the other clusters, e.g.,  $D(I_i, C_g) = \min(D(I_i, C_1), \dots, D(I_i, C_k))$ . Create a new cluster  $C_{k+1}$  and assign  $I_i$  to it, i.e.,  $C_{k+1} \leftarrow I_i$ , if  $I_i$ 's maximum distance from all the clusters is greater than the average of all the distances from different clusters, e.g.,  $\max(D(I_i, C_1), \dots, D(I_i, C_k)) > \text{avg}(D(I_i, C_1), \dots, D(I_i, C_k))$ .

The random selection of individuals while creating groups can enhance diversification. However, sequential and distance-based clustering methods can provide better intensification. The fitness-proportionate grouping process also maintains a good amount of diversification. A comparison of all the social group-forming mechanisms in terms of intensification and diversification is presented in Table 3.

- B. *Evaluation*: Evaluation of individuals is typically performed to find the leaders and create groups of individuals. A grouping mechanism may utilize the individuals' evaluated fitness to create highly diverse groups or better-resembling groups in terms of their fitness. However, some grouping mechanisms, the distance-based clustering method, do not require prior evaluation of the individuals. Thus, the evaluation of individuals may be performed in one of two ways as follows.
  - (a) Individuals are evaluated (typically to determine the leaders), and then grouping is performed.
  - (b) Individuals are evaluated after their grouping.
- C. *Leader Selection*: Some grouping mechanisms utilize selected leaders to create groups, whereas some do not require leader selection. However, leaders can be generated beforehand, and the remaining population can be generated in the proximity of these generated leaders. Thus, leader selection is performed in one of the following ways.
  - (a) The best of the evaluated individuals are selected as leaders, and groups are created by allocating the remaining individuals sequentially or randomly to these leaders.
  - (b) Groups are created using some distance-based method. Then the best of the evaluated individuals are announced as leaders.
  - (c) Leaders are generated randomly in the initial phase of an algorithm. Then group mem-

bers are generated in the leader's neighborhood. After evaluating all group members, leaders are updated by the best individuals in the group. This selection of leaders is typically based on their objective function values. However, it may also consider constraint satisfaction in the case of constrained optimization problems.

A schematic of the above-described grouping mechanisms is provided in Fig. 3 to differentiate between them. Each downward trail determines a different grouping mechanism in this figure.

2. *Reproduction Mechanisms*: A reproductive mechanism used in socially motivated systems depicts the social convention of individual evolution. Imitation and transfer of mannerisms are suggested and utilized by various SIEAs as reproductive mechanisms. These reproduction mechanisms mostly exhibit the Lamarckian evolution of individuals, i.e., acquired improvements in an individual's fitness will change its genetic encoding rather than just updating its fitness or creating a new individual. As the best individual in a group is the most influential entity, other individuals in the group mimic the characteristics of this best individual to evolve themselves. Similarly, the best global individual influences the group leaders to follow him. Moreover, individuals can follow other better peers or use self-improvement methods. These reproductive mechanisms typically simulate asexual versions of various recombination mechanisms used in the Darwinian evolutionary system. Hence, the reproduction in SIEAs can occur in two phases.

- A. *Intra-group Reproduction*: Intra-group reproduction mechanisms utilized by SIEAs are as follows.
  - (a) Most SIEAs use the "Follow the Leader(s)" principle within a group. The best group individual(s) influences other ingroup members in this phase. Group individuals are usually modified to become more like the leader(s) or are moved toward the leader(s).
  - (b) The individuals follow better peer individual(s) to achieve a better place in the group.
  - (c) Random changes in the population members are made to enhance or maintain population diversity.
  - (d) New randomly generated members are introduced to groups. These new individuals usually replace the worst members of the group.
- B. *Inter-group Reproduction*: The inter-group reproduction can happen in multiple ways.
  - (a) The global best individual influences the best individuals (local leaders) from all other groups in this phase, i.e., each group's best individual is usually moved toward the global best. This

movement may be produced using appropriate operators or the same method used for intra-group reproduction.

- (b) The local leaders follow another leader(s).
- (c) The local leaders go through self-observation and introduce a quantum of change in themselves using mutation-like operators.
- (d) Individuals interact and follow the members of other groups.
- (e) Individuals switch their groups.

Following leaders or members can be modeled with a reinsertion operator or a weighted movement of members toward leaders or other members for increased search power that contributes to diversification in the initial phases and intensification toward the end of the algorithm. The quantum of change in each group member is essential in deciding how fast the member converges toward the leader(s) or other member(s). Too rapid convergence may destroy diversity in the group, resulting in a local optimum. However, too slight changes increase computational cost. As leaders change themselves in the inter-group reproduction process, members of the population that were hitherto closer in search space to one leader may become close to another leader and change their group affiliation accordingly. A new randomly generated individual may replace a leader or member exhibiting no improvement for a predefined number of iterations. Likewise, replacing the weak members of a group with new randomly generated individuals can promote diversification. The intra- and inter-group reproduction phases can be implemented in an intertwined or reverse sequence manner. Initially, the groups must explore the subpopulation with a higher diversification rate. So in the initial stages, the algorithm typically performs more exploration. As the algorithm progresses, the individuals converge toward their respective leaders, focusing more on intensification.

3. *Group Refreshment*: Groups in SIEAs are updated after or between each iteration of intra- and inter-group reproduction processes. This refreshment of groups may be performed using one or more of the following.

- (a) To maintain the influence cycle in each group, the group leaders are updated, i.e., the group member(s) exhibiting better-evaluated value (fitness) than the leader(s) exchanges its (their) role with it (them). As a result, the global best individual is also updated.
- (b) The low-powered groups are merged into other strong groups (influenced by their possessiveness or strength).
- (c) Strong groups are merged to form a diverse group with better individuals.

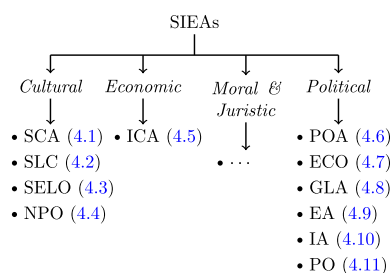


Fig. 4 Classification of SIEAs

- (d) Despite the resource-intensive and expensive nature of the grouping process, some SIEAs undergo regrouping after each iteration for a fresh start, i.e., the groups are reformed with the updated population after intra- and inter-group processes.

4. *Termination Criterion*: The following conditions can be utilized as the termination condition for an SIEA.

- (a) The population in an SIEA is diverted toward the global optimum solution after each group refreshment, which may include the migration of group members to other groups and the collapsing of low- or no-power groups into other better groups. Often, only one group remains after many group refreshments; in that case, SIEA ends.
- (b) SIEA terminates after a predefined number of iterations, a standard termination criterion in EAs.
- (c) Due to the higher convergence speed of an SIEA, the population within a group may become similar in a relatively short period. In such cases, the termination criteria in SIEAs can be modeled in terms of the comparative similarity of the whole population. If the population has not changed significantly for several iterations, there is no need to run the algorithm anymore; hence, a termination point is reached.

## 4 Prior art on SIEAs

In this section, eleven well-known SIEAs are reviewed in the order of their appearance in the classification as in Table 2 and Fig. 4. For each SIEA, inspirational social phenomena, their algorithmic description, and pseudocode are presented. Furthermore, a table of algorithmic steps utilized in each SIEA in reference to the proposed general framework is provided. These tables highlight the differences between the algorithmic steps used by different SIEAs. Also, a consolidated table combining these tables is furnished at the end of the section to provide a broad view of the similarities and differences between reviewed SIEAs.



## 4.1 Society and civilization algorithm (SCA)

Society and civilization algorithm (Ray and Liew 2003) is inspired by the human behavior of interacting in a symbiotic relationship and improving while living together in societies. The algorithm rests on the following basic premises.

Social interactions are improvement-oriented, i.e., society members interact to create opportunities to improve themselves. A society's success relies on its members' progress and success. Hence, a good balance of cooperative and competitive relationships among individuals advances societies and civilizations. Other social phenomena in societies include migration, knowledge sharing, and leadership.

The algorithmic steps for SCA are outlined as follows. And the correspondence of these steps with those of the general framework is given in Table 4.

1. *Initialization and groups formation*: A group of individuals forms a society, and societies form a civilization. In SCA, some random points (individuals) are initially generated in the parametric space. These generated individuals together represent a complete civilization. All individuals in the civilization are evaluated for two types of fitnesses corresponding to the objective function and constraint satisfaction. Individuals close to each other in terms of Euclidean distance are assigned to the same society. Thus, mutually exclusive but collectively exhaustive societies are created so that every society has some individuals, and every individual of the civilization is assigned to some society. The best individuals in terms of both fitnesses in each society are designated leaders. Likewise, the best leaders become civilization leaders.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: Society leaders facilitate the improvement of other society individuals. The movement of individuals toward nearby society leaders in the parametric space achieves this. In effect, this simulates knowledge acquisition from the leader to improve individuals. In terms of EAs, this process implements intensification.
  - (b) *Inter-group reproduction(s)*: Society leaders improve by migrating toward the civilization leaders. The migration of society leaders expands promising regions in parametric space by receiving increased individual counts. In terms of EAs, this movement implements diversification. The newly generated improved individuals by intra- and inter-group improvement processes and the civilization leaders directly appear in the subsequent civilizations.
3. *Group refreshment*: Society formation and leader selection are repeated at each subsequent iteration.

**Table 4** SCA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in SCA
1A. Grouping mechanism	vi
1B. Evaluation	(a)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(a)
2B. Inter-group reproduction(s)	(a)
3. Group refreshment	(d)
4. Termination criteria	(b)

4. *Termination condition(s)*: The termination of SCA depends on the computational resources available and is specified as the count of maximum iterations permitted, i.e., the number of civilizations counts. SCA stores parametrically unique individuals (solutions) across the civilizations, enabling diverse solutions to be maintained in the search process. At termination, the civilization leaders are reported as the best-found solutions.

SCA was initially proposed for constrained optimization problems. In SCA, for each individual, a constraint satisfaction vector is calculated and maintained throughout the run. A nonzero value at the  $i^{\text{th}}$  position in this vector represents  $i^{\text{th}}$  constraint violation, and a zero at the  $i^{\text{th}}$  position means  $i^{\text{th}}$  constraint satisfaction. Pseudocode for SCA is provided in Algorithm 2.

### Algorithm 2 Society and Civilization Algorithm

- 1: Initialize the parameters and generate a random population representing a civilization
- 2: Evaluate each individual according to the objective function and constraint satisfaction
- 3: **repeat**
- 4:   Build mutually exclusive societies using the clustering method
- 5:   Identify leaders and create a leader set for each society
- 6:   Move each non-leader individual in the direction of the nearest leader and add this newly positioned individual to the new civilization
- 7:   Identify the civilization leader
- 8:   Move all the leaders except the civilization leader in the direction of the nearest best leader and add these newly positioned leaders to the new civilization
- 9:   Add the civilization leader to the new civilization
- 10: **until** given civilization count

Talent-based social algorithm (Daneshyari and Yen 2004) is an improved version of SCA. This algorithm proposes two new Liberty Rate and Talent concepts to provide new mechanisms for moving individuals toward corresponding leaders. Liberty Rate measures society's individuals' independence, defined as the ratio of society's average fitness and the aver-

**Table 5** Hybrid versions of SCA

Hybrid Version	Application	Publications
SCA + Shuffled Frog Leaping Algorithm (SFL) + GA	Traveling salesman problems	Bonyadi et al. (2007); Eusuff and Lansey (2003)
SCA + Particle Swarm Optimization (PSO)	Economic load dispatch problems	Selvakumar and Thanushkodi (2009); Kennedy and Eberhart (1995)
SCA + Gravitational Search Algorithm (GSA)	Single and multiple distributed generation placement problems in distribution networks	Hosseini-Hemati et al. (2021); Rashedi and Nezamabadi-Pour (2007)
SCA + Variable Neighborhood Search (VNS)	Parallel-machine serial-batching scheduling problems	Pei et al. (2021); Mladenović and Hansen (1997)

age fitness of civilization. Liberal societies allow people to move freely toward the leaders. Moreover, each individual's Talent is defined as the product of its objective and constraint ranks.

Civilized Swarm Optimization (CSO) (Selvakumar and Thanushkodi 2009), a prominent hybrid variation of SCA, integrates the self-experience concept from particle swarm optimization (PSO). In CSO, a society individual explores society through guidance based on their own experience and that of their leaders. Table 5 provides a list of hybrid versions of SCA.

## 4.2 Soccer league competition algorithm (SLC)

Soccer league competition algorithm (Moosavian and Roodsari 2013) is inspired by the competitions within and among soccer teams seen in soccer leagues. In a soccer league, players compete for a better position. In contrast, the teams compete for a better rank at the league table. Four socio-inspired operators—imitation, provocation, self-inspection, and substitution are utilized in SLC. The imitation operator expedites the algorithm's searching capability, while the provocation operator provides high-accuracy solutions to complex optimization problems. However, mutation and substitution operators help the algorithm escape from local minima and plateaus. The algorithm rests on the following basic premises.

Every team in a league has several fixed and substitute players. Super player (SP) and super-star player (SSP) represent the best player in a team and among all teams, respectively. Team players admire and follow their SP or other team's SP, while each SP admires and follows the SSP. After each match, the winner and the loser are declared, and some players, including fixed and substitutes, experience changes. These changes aim to improve the performance of both players and teams. The winning team players go through imitation and provocation processes, whereas the losing team members self-inspect their strategies (using the self-inspection operator) and are substituted more often (using the

substitution operator). In imitation, fixed players in a winning team imitate both the SP and the league SSP to improve their future activities. In provocation, substitute players are promoted to fixed players when their performance exceeds the average performance level of the fixed team players. In Self-inspection, a fixed player in a loser team revises his activity to prevent failure in future games. For this, that player tries other strategies that may benefit him. In Substitution, new substitutes are tested in a match for their performance.

The algorithmic steps for SLC are outlined as follows. The correspondence of the steps of SLC with those of the general framework is provided in Table 6.

1. *Initialization and groups formation*: The initial population in SLC represents the complete set of players in a league. These players are evaluated for their fitness, called their power, and are sorted to form some teams. There are two types of players in a team—fixed players and substitutes, and the team's total power is defined as the average power of these players. When a league starts, competitions between all possible pairs of teams are performed, and the winning and losing teams are declared based on their power. In a league, teams compete for a higher rank in the league table, and players compete for a higher position in their team. The best-ranked player in a team is super player (SP), and the best-ranked player in the league is super-star player (SSP).
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: In the winning team, fixed players imitate their team SP, and substitute players are promoted to fixed players if their power exceeds the team's mean power. While in the loser team, fixed players self-inspect their behavior and try mutation to improve their strategy, and new substitutes are introduced for a change.
  - (b) *Inter-group reproduction(s)*: Fixed players of the winning team imitate the SSP of the league.

**Table 6** SLC and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in SLC
1A. Grouping mechanism	i(a)
1B. Evaluation	(a)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(a), (c) & (d)
2B. Inter-group reproduction(s)	(d)
3. Group refreshment	(d)
4. Termination criteria	(b)

3. *Group refreshment*: After performing operators on winning and losing teams, all the players are sorted if the termination condition is not satisfied, and new teams are formed for the successive leagues. Again, the power of competing players and teams in the league is calculated. Furthermore, SP and SSP are also updated.
4. *Termination condition(s)*: The number of seasons is the termination condition in this algorithm, and the SSP at the termination represents the best-found global solution.

The pseudocode for SLC is provided in Algorithm 3.

---

**Algorithm 3** Soccer League Competition Algorithm
 

---

- 1: Initialize the parameters and generate random players
  - 2: Evaluate players' fitness
  - 3: **repeat**
  - 4:   Form the teams and allocate sorted players to them sequentially
  - 5:   Start the league competition
  - 6:   Find the losing and winning teams
  - 7:   Apply corresponding operators in both the teams
  - 8: **until** maximum number of seasons
- 

Diversity-team soccer league competition algorithm (DSLCL) (Qiao et al. 2020) is an improved version of SLC. It improves SLC by adding trading and drafting of players and combining these strategies. In trading, players between teams are exchanged, while new players are introduced to a team in drafting. Table 7 lists a hybrid version of SLC.

### 4.3 Socio evolution and learning optimization (SELO)

Socio evolution & learning optimization algorithm (Kumar et al. 2018) is inspired by human learning behavior as a member of the modern societal system known as family. A family is an elementary social group where people live closely and follow some communication protocols continuously all day. These communications yield the influence of the better members on the other members, and SELO imitates this influential

process seen in families to improve the members' behavior. The algorithm rests on the following basic premises.

Each family member possesses different behavioral traits, which result from successive observation and imitation of the behavior of other family members. Children learn values, behaviorism, and manners from their parents, peers, and other members of society. Moreover, every society individual learns mannerisms and behaviors from other individuals, e.g., parents learn parenting and other things from other parents. This interaction and learning from others helps in improving individual performance.

Below are the algorithmic steps for SELO. Table 8 provides the correlation of these steps with those of the general framework. The pseudocode for SELO is provided in Algorithm 4.

1. *Initialization and groups formation*: Initially, several families with an equal number of individuals (2 parents and a few children) are initialized. Individuals in these families are generated in their close neighborhood in the objective space using a clustering mechanism. Firstly parents are generated in a close neighborhood, and then kids are generated following one of the parents. These families collectively represent a society. Each family's parents and kids are evaluated for their fitness, and the global best society member is determined.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: After initialization, each member in a family follows another better member selected with a roulette-wheel selection procedure. Parents try to follow other better parents, and kids follow their parents and better siblings. If following other better parents in society does not improve a parent, then a self-contemplation operator (modeled by mutation operation) is utilized.
  - (b) *Inter-group reproduction(s)*: Kids in a family follow their parents in the initial phase. Gradually this influence is shifted to their siblings, peers, and finally, to other better society individuals. Similarly, parents follow parents from other families. Whenever a kid is influenced to a wrong path, i.e., the newly generated solution is worse than the current solution, parent intervention is required; a behavior correction operator is utilized to rectify the situation.
3. *Group refreshment*: Global best society member is re-determined at the beginning of each iteration.
4. *Termination condition(s)*: The algorithm terminates when the maximum number of learning attempts are performed, or families converge. The convergence of a family represents the saturation of all the family members in a single position. At termination, any of these can be reported as

**Table 7** Hybrid Version of SLC

Hybrid Version	Application	Publications
SLC + Pattern Search	Optimum sitting and sizing of wind turbines (WTs), Photovoltaics (PVs), Energy Storage Systems (ESSs), and Electric Vehicles Charging Stations (EVCSS)	Dogan (2021b)

**Table 8** SELO and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in SELO
1A. Grouping mechanism	iv(a)
1B. Evaluation	(b)
1C. Leader selection	(c)
2A. Intra-group reproduction(s)	(a), (b) & (c)
2B. Inter-group reproduction(s)	(a), (b) & (d)
3. Group refreshment	(a)
4. Termination criteria	(b)

**Table 9** NPO and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in NPO
1A. Grouping mechanism	iv(a)
1B. Evaluation	(b)
1C. Leader selection	(c)
2A. Intra-group reproduction(s)	(c)
2B. Inter-group reproduction(s)	(a)
3. Group refreshment	(a)
4. Termination criteria	(b)

the best solution found. As a rule, the best family in society contains the best solution(s).

surrounding regions to find suitable locations. Whenever a better place is located, Sheikh moves the entire clan there.

The algorithmic steps for NPO are described below and mapped to those of the general framework in Table 9.

#### Algorithm 4 Socio Evolution and Learning Optimization Algorithm

- 1: Initialize the parameters and generate the population of parents in their pairwise close neighborhood
- 2: Generate kids in the neighborhood of the parents in a family
- 3: **repeat**
- 4: Find the best family and society member
- 5: Each parent of every family decides to randomly follow the behavior of a parent from one of the other families
- 6: Kids decide to follow either their parents or their siblings or kids from other families
- 7: **until** maximum number of learning attempts, or families converge

#### 4.4 Nomadic people optimizer (NPO)

Nomadic people optimizer (Salih and Alsewari 2020) is inspired by the behavior of nomads in the search for required food and water. NPO is motivated by the lifestyle of Bedouin Arabic nomads known as Bedu or Beduin. The algorithm rests on the following basic premises.

Bedouins travel their entire life with all their belongings (mostly camel, cattle, and sheep herds) in search of locations rich with necessary resources for their lives. Their families are classified as normal families and the Sheikh family. Sheikh represents the clan leader and determines the places for the families' survival and distribution pattern at the location. The Sheikh selects a few normal families to explore the

1. *Initialization and groups formation*: Initially, a set of clan leaders is generated randomly. Then, other families are generated in the vicinity of these clan leaders. Leaders and other families are evaluated for their fitness, and if any new family is found to be better than the previous leader, the leader for that clan is updated.

#### 2. *Reproduction mechanisms*:

- (a) *Intra-group reproduction(s)*: If no newly generated family is better than the corresponding clan leader, then the families use levy flight (Kamaruzaman et al. 2013) moves to search for better locations in other regions. Again, after performing levy flights, clan leaders are updated.
- (b) *Inter-group reproduction(s)*: The global best leader is determined among the leaders, and all other leaders strive to follow that leader.

3. *Group refreshment*: Group refreshment is applied after each intra-group improvement. Leaders in each clan are updated. Furthermore, the best global leader is determined at the beginning of inter-group improvement process.
4. *Termination condition(s)*: NPO terminates after a maximum number of iterations have been completed.

The pseudocode for NPO is provided in Algorithm 5, and Table 10 lists its hybrid version.

**Algorithm 5** Nomadic People Optimizer

- 1: Initialize the parameters and generate the leaders or Sheikh
- 2: Evaluate the leaders
- 3: **repeat**
- 4:   Generate normal families in the proximity circle of the leaders and evaluate them
- 5:   Update the leaders if newly generated normal families have better fitness than their corresponding clan leader; otherwise, explore the search space using levy flight
- 6:   Update the leaders with the families having better fitness than their corresponding leaders after levy flight
- 7:   Periodical Meetings: leaders participate in these meetings aimed to provide the best leader and move all the normal leaders toward the best leader
- 8: **until** maximum number of iterations

**Table 10** Hybrid Version of NPO

Hybrid Version	Application	Publications
Principal Component Analysis (PCA) + NPO + Gradient Boosting Decision Tree (GBDT)	Microgrid fault detection and identification	Gopinath and Balakrishnan (2022)

**4.5 Imperialist competitive algorithm (ICA)**

Imperialist competitive algorithm or colonial competitive algorithm (CCA) (Atashpaz-Gargari and Lucas 2007b; Atashpaz Gargari et al. 2008) is inspired by imperialism and the imperialistic competition among empires to improve themselves by taking control of weaker colonies of other empires. The algorithm rests on the following basic premises.

Stronger countries overcome weaker countries and form their empires. The strongest country in each empire is called imperialist, and the possessed countries are called colonies of these imperialists. Imperialists attempt to improve colonies by imposing their more profitable characteristics upon them. Colonies can revolute during this process and can exhibit better characteristics independently. Stronger empires acquire colonies from weaker empires through imperialistic competition and become more powerful. An empire with no colonies suffers collapse or is defeated by a better/best empire in this imperialistic competition.

Below are the algorithmic steps for ICA. Table 11 provides the correlation of these steps with those of the general framework. And the pseudocode for ICA is provided in Algorithm 6.

1. *Initialization and groups formation*: A random population of countries (solutions) is generated in the parametric space. Each gene in such countries represents one characteristic. These countries are evaluated for their cost (fitness). A few best countries are designated imperialists, and the rest are termed colonies. The imperialists

**Table 11** ICA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in ICA
1A. Grouping mechanism	iii(b)
1B. Evaluation	(a)
1C. Leader selection	(a)
2A. Intra-group reproduction(s)	(a) & (c)
2B. Inter-group reproduction(s)	(e)
3. Group refreshment	(a) & (b)
4. Termination criteria	(a) & (b)

overcome nearby weaker countries in proportion to their cost to form empires.

2. *Reproduction mechanisms*:

(a) *Intra-group reproduction(s)*: Imperialists impose their characteristics on the possessed colonies. Colonies move in objective space toward their imperialist to simulate this process. In effect, this imitates the knowledge acquisition from the imperialists to the colonies for their improvement. In terms of EAs, this implements exploration. Colonies occasionally undergo some revolution while following imperialists for exploitation. In this moving process, a colony may attain better fitness than its imperialist. Then, the roles of that colony and its imperialist are interchanged.

(b) *Inter-group reproduction(s)*: Empires compete to acquire the weakest colony of the worst empire. Thus, better empires flourish, weaker ones weaken, and, finally, collapse into better ones.

3. *Group refreshment*: Imperialists in each empire represent the best solution in that empire and are updated after performing assimilation and revolution processes. Furthermore, empires with no colonies collapse into strong empires after the imperialistic competition.
4. *Termination condition(s)*: The weaker colonies and empires collapse into stronger empires, and eventually, the algorithm ends up with just one empire. In this case, the imperialist of the only empire is reported as the best solution found, and the algorithm is terminated. Additionally, or alternatively, the maximum iteration count can be specified as the termination condition. Empires are distinguished based on their power, calculated using the cost of their imperialists and colonies. Upon termination, the imperialist of the best empire provides the best solution.

Social-based algorithm (Ramezani and Lotfi 2013) provides a hybrid version of ICA by combining EA and ICA. Various discrete versions of ICA have also been proposed over time,

**Algorithm 6** Imperialist Competitive Algorithm

- 1: Initialize the parameters and generate some random solutions (countries)
- 2: Create the empires
- 3: **repeat**
- 4: Assimilate and revolute empires: move the colonies toward nearby best imperialist with some probability
- 5: Exchange colony and imperialist, if the colony's fitness is better than imperialist
- 6: Compute the total power of empires
- 7: Imperialist Competition: Move the weakest colony (colonies) from the weakest empire to the empire that has the most likelihood of possessing it
- 8: Eliminate empires having no colonies
- 9: **until** only one empire remains

**Table 12** Hybrid Versions of ICA

Hybrid Version	Application	Publications
ICA + EA	Global optimization	Ramezani and Lotfi (2013)
ICA + PSO	Optimal reactive power dispatch of power systems	Mehdinejad et al. (2016)
ICA + ABC	Sentiment classification	Osmani et al. (2022)

viz. discrete binary ICA (DB-ICA) Nozarian et al. (2012), discrete ICA (DICA) Emami and Lotfi (2013), modified ICA (MICA) Mousavirad and Ebrahimpour-Komleh (2013), and binary ICA (BICA) Mirhosseini and Nezamabadi-pour (2018) to name a few. In a survey of industrial engineering applications of ICA, it was found that there is an increased volume of research being published in this area (Hosseini and Khaled 2014). The broad applicability of ICA motivates an independent review of its improved and hybrid versions. Some hybrid versions of ICA are listed in Table 12.

#### 4.6 Parliamentary optimization algorithm (POA)

Parliamentary optimization algorithm (POA) (Borji 2007; Borji and Hamidi 2009) is inspired by human competitive behavior seen in the parliamentary head selection process. This algorithm also mimics human behavior in an athletic championship or presidential campaign competition. The algorithm rests on the following basic premises.

The members of parliament belong to a political party and are elected in a general election process. These parliament members of a political group are divided into candidates and regular members in a parliamentary head election. Candidates are the members nominated for the parliamentary head position. Meanwhile, regular members vote for the candidates according to their interests. They can migrate to other parties influenced by their ideology. In parliamentary elections, candidates or nominated party leaders try to get as

many votes as possible from their party members. At times political parties also form alliances to win the elections.

The algorithmic steps for POA are as follows. The correspondence of these steps with those of the general framework is provided in Table 13. Further, the pseudocode for POA is provided in Algorithm 7.

1. *Initialization and groups formation*: A random population of individuals is generated, each representing a member of parliament. This population of members is divided randomly into a few equal-sized political groups or parties. Each parliament member is marked as either a candidate or a regular member depending upon their evaluated fitness. As the representatives of their respective parties in a parliamentary head election, the best party members are selected as candidates. A regular member supports them by voting in their favor.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: Regular political party members favor candidates to improve their fitness. If a regular member attains better fitness than a candidate, their roles in that party are exchanged.
  - (b) *Inter-group reproduction(s)*: The overall power of each political group is calculated in terms of its members' fitness. Members of weaker groups are forced to change their parties to better political groups. In this process, the worst-performing groups gradually lose their power and eventually collapse into other groups. Alliances are also formed with a predefined probability.
3. *Group refreshment*: Candidate members in each group are reassigned after intra-group improvement. Some of the best groups are merged, and some worst groups are deleted with some predefined probabilities.
4. *Termination condition(s)*: Unlike the real world, POA converges to a state where only one political party remains. The best solutions in POA are represented as candidates in each political party. The best among them is the best solution in each iteration.

An improved version of POA for permutation constraint satisfaction problems is proposed by de Marcos et al. (2010). Table 14 lists some hybrid versions of POA.

#### 4.7 Election campaign optimization (ECO)

Election campaign optimization (ECO) algorithm (Lv et al. 2010) simulates the candidates' behavior seen in an imaginary election campaign scenario to achieve the highest support from the voters. The algorithm rests on the following basic premises.

**Table 13** POA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in POA
1A. Grouping mechanism	i(a)
1B. Evaluation	(b)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(a)
2B. Inter-group reproduction(s)	(d)
3. Group refreshment	(a) & (c)
4. Termination criteria	(a)

**Table 14** Hybrid Versions of POA

Hybrid Version	Application	Publications
POA + ANN	Passenger flow prediction	Pekel and Soner Kara (2017)
POA + BB-BC	CEC 2005 mathematical functions	Kiziloluk and Özer (2019)

**Algorithm 7** Parliamentary Optimization Algorithm

- 1: Initialize the parameters and generate a random population
- 2: Divide the population into some equal-sized groups
- 3: Evaluate the individuals
- 4: Select the best fit individuals in each group as candidate members
- 5: **repeat**
- 6:   Bias regular members of each group toward the candidates of the same group
- 7:   Reassign candidate members
- 8:   Compute each group’s power
- 9:   Select some best groups and merge them with some predefined probability
- 10:   Delete some groups with a predefined probability
- 11: **until** only one political group remains

Candidates with a higher social status (prestige) can participate in elections. Candidates organize campaigns to get maximum support from the voters for their election advancement. Voters support candidates that they find promising for their betterment.

The algorithmic steps for ECO are outlined as follows. The correspondence of these steps with those of the general framework is provided in Table 15. Further, the pseudocode for ECO is provided in Algorithm 8.

1. *Initialization and groups formation*: The population space in this algorithm comprises two types of individuals—candidates and voters. Initially, a random population of candidates is generated. Then, voters are generated globally using uniform distribution and into the candidates’ support regions (ranges) using the normal distribution. The support regions of candidates represent the area in which candidates can affect voters. The candidate’s effect decreases as the difference between the candidate and the

**Table 15** ECO and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in ECO
1A. Grouping mechanism	iv(a)
1B. Evaluation	(b)
1C. Leader selection	(a)
2A. Intra-group reproduction(s)	(a)
2B. Inter-group reproduction(s)	(d)
3. Group refreshment	(a)
4. Termination criteria	(b)

voter increases. After a range limit of support region, it reduces to zero. The support regions can overlap, i.e., a voter can be affected by one or more candidates depending on their support ranges.

2. *Reproduction mechanisms*:

- (a) *Intra-group reproduction(s)*: Voters affected by multiple candidates distribute their support in the proportion of their effect. In this process, a voter’s prestige can become better than a candidate’s. Then, the voter’s rule is interchanged with the candidate.
- (b) *Inter-group reproduction(s)*: Through campaigns, candidates affect voters in their support regions. In return, voters support candidates in proportion to their prestige. The voters’ support helps the candidate change to an updated location in the next iteration, which defines its new election location. A candidate’s total support is the aggregated sum of all the affected voters.

3. *Group refreshment*: Candidates are reassigned in each iteration with the voters with better prestige in the same support region.

4. *Termination condition(s)*: A maximum number of iterations is used as the termination condition. At termination, the candidate with the highest prestige provides the best global solution.

**Algorithm 8** Election Campaign Optimization

- 1: Initialize the parameters and generate a random population of candidates
- 2: Calculate candidates’ prestige and ranges
- 3: Generate voters
- 4: **repeat**
- 5:   Calculate the effect of candidates on voters
- 6:   Calculate the prestige and support of the voters
- 7:   Find the new support focus and ranges of the candidates
- 8:   Substitute the candidate with the voter with better prestige than the candidate if any
- 9: **until** maximum number of iterations

A comparative study verifying the good performance of ECO on constrained optimization problems is reported by Xie et al. (2010). A parameter design and performance study on the ECO algorithm is presented by Zhang et al. (2011b).

#### 4.8 Group leaders optimization algorithm (GLOA or GLA)

Group leaders optimization algorithm (Daskin and Kais 2011b) is inspired by the influence people get from the leaders and the members of other groups. The algorithm rests on the following basic premises.

Leaders work as the inspirational body in a group and are the best individuals. Leaders influence other individuals and inspire them to improve. Individuals also get inspiration from other better members of different groups. These influenced members can replace the leader if they attain better characteristics than the leader.

The algorithmic steps of GLOA are as follows. Table 16 shows how they correspond to those of the general framework. And the pseudocode for GLOA is provided in Algorithm 9.

1. *Initialization and groups formation*: In this algorithm, a few groups are generated randomly, and then these groups are inflated with an equal-sized population of randomly generated members. Each member of these groups is evaluated for fitness, and the best-fit member in each group is designated as a leader.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: The leader in each group helps improve individuals using recombination and mutation operators. Randomly selected members in each group are crossed-over with the group leader, and the best between the previous and newly generated members is stored.
  - (b) *Inter-group reproduction(s)*: One-way crossover is performed between different groups to maintain diversity in each group. In this process, randomly selected members from one group influence randomly selected individuals in each group. Characteristics of one member are overwritten on another member, and the best between the previous and newly generated members is stored.
3. *Group refreshment*: Sometimes, a newly generated group member surpasses its leader; hence, the leader is redetermined at the start of each iteration.
4. *Termination condition(s)*: The maximum number of iterations is used to terminate the algorithm. The leader in each group represents the best local solution. The best of these leaders represents the best-found solution.

**Table 16** GLOA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in GLOA
1A. Grouping mechanism	i(a)
1B. Evaluation	(b)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(a) & (c)
2B. Inter-group reproduction(s)	(d)
3. Group refreshment	(a)
4. Termination criteria	(b)

**Table 17** Hybrid Versions of GLOA

Hybrid Version	Application	Publications
Quantum Evolution Algorithm + GLOA + Chaotic Method	Green material selection with energy-consideration (GMS-EC) in product design	Tao et al. (2016)
Local search + Differential Evolution + GLOA	Energy-Aware Material Selection	Bi et al. (2017)
Dimensional self-adaptation + GLOA + Artificial bee colony algorithm	Logistics-aware manufacturing service collaboration optimization	Wang et al. (2019)

#### Algorithm 9 Group Leaders Optimization Algorithm

- 1: Initialize the parameters and generate a random population of individuals
- 2: Create equal-sized groups
- 3: Evaluate the individuals
- 4: **repeat**
- 5:   Determine the leader for each group
- 6:   Generate new individuals using the crossover operator between an individual and the group leader and mutate the generated individual
- 7:   Influence individuals from other groups individuals selected randomly
- 8: **until** maximum number of iterations

Xiang et al. (2014) proposed a Pareto-GLA to solve multi-objective optimization algorithms. Table 17 lists some hybrid versions of GLOA.

#### 4.9 Election algorithm (EA)

The election algorithm (Emami and Derakhshan 2015) is inspired by the presidential elections used to elect a president in a country. This algorithm mimics the election strategy used in the real world. The algorithm rests on the following basic premises.

Every individual participates in an election, either as a candidate or as a voter. Initially, candidates and their supporters



form political parties. These supporters follow their party candidate because of his ideology. Candidates start advertising campaigns to attract more and more supporters in their favor. This advertisement can be of two types—positive and negative advertisement. In a positive advertisement, candidates convey their good characteristics, while in a negative advertisement, candidates disparage their opponents. During advertising campaigns, candidates with the same ideologies might join and create a joint party. This coalition of parties increases their chance of success in the election.

The algorithmic steps for EA are outlined as follows, and the correspondence of these steps with those of the general framework is provided in Table 18. Algorithm 10 provides the pseudocode for the election algorithm.

1. *Initialization and groups formation*: In the election algorithm, a random population of individuals is generated. These individuals can be of one of two types—a candidate or a voter (or supporter). Initially, a few individuals are selected as candidates from this population. Afterward, using the clustering method, various electoral parties are formed according to the same interests, beliefs, and ideas to participate in the election.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: Advertising is the counterpart of operators in GA. Before the election starts, electoral parties try to influence individuals using advertising. The election algorithm has three ways of advertising: positive advertisement, negative advertisement, and coalition. In a positive advertisement, candidates impact the individuals by introducing their positive images and qualities.
  - (b) *Inter-group reproduction(s)*: In a negative advertisement, candidates impact individuals by introducing negative images and qualities of their opponents. This advertisement affects individuals globally and converges them to a specific electoral party. This winning electoral party contains the global optimum of the whole solution space.
3. *Group refreshment*: Candidates at the end of each iteration are reassigned for each party. Furthermore, candidates with similar ideas join to create a united party in the coalition process.
4. *Termination condition(s)*: An election day is used as the termination criterion, defined as the maximum number of iterations. The candidates in each group represent the best solution in the group. The best of these candidates represent the best-found solution.

A new election algorithm based on assistance in distributed systems (Zargarnataj 2007) is an improvement over this algorithm. Emami (2019) proposed an improved election algo-

**Table 18** EA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in EA
1A. Grouping mechanism	v
1B. Evaluation	(b)
1C. Leader selection	(a)
2A. Intra-group reproduction(s)	(a)
2B. Inter-group reproduction(s)	(d)
3. Group refreshment	(a) & (c)
4. Termination criteria	(b)

**Algorithm 10** Election Algorithm

- 1: Initialize the parameters and generate an initial population of individuals
- 2: Evaluate each individual
- 3: Create electoral parties
- 4: **repeat**
- 5: Candidates advertise their plans and enhance their positions within the party by learning new ideas
- 6: Candidates aim to win over supporters from other parties
- 7: Collate candidates if they have similar ideas
- 8: Re-evaluate the eligibility of candidates
- 9: **until** maximum number of iterations

rithm by modifying the party formation step and introducing a chaotic positive advertisement and migration operator. A modified version of the election algorithm to solve the random  $k$  satisfiability problem is provided by Abubakar et al. (2020); Sathasivam et al. (2020).

A comparison between the election and election campaign optimization algorithms is presented in (Abubakar and Sathasivam 2020). Abubakar et al. (2020) reported a significant enhancement in the election algorithm on incorporating a negative campaign strategy. Table 19 lists the hybrid versions of the election algorithm.

**4.10 Ideology algorithm (IA)**

Ideology algorithm (IA) (Huan et al. 2017) is a socio-politically inspired population-based meta-heuristic algorithm. It simulates political party members’ self-interest and competitive behavior to improve their rank in the party. The algorithm rests on the following basic premises.

Ideologies guide individuals to achieve their life goals. Hence, individuals in a society follow and support some ideologies. In the case of local parties, each party follows certain ideologies. The individuals in a party follow their party’s ideology. At the same time, every party member wants to become the local party leader, and every local party leader wants to become a global leader.

The algorithmic steps for IA are outlined as follows. The correspondence of the steps of IA with those of the general

**Table 19** Hybrid Versions of EA

Hybrid Version	Application	Publications
EA + Kriging Method	Groundwater monitoring network design	Kavusi et al. (2020)
Artificial Neural Network + EA	Random Boolean $k$ Satisfiability	Abubakar and Danrimi (2021)
EA + Support vector regression method (SVR)	Discharge coefficient (Cd) estimation of vertically cosine shape weirs	Emami et al. (2022)

framework is provided in Table 20. The pseudocode for IA is provided in Algorithm 11.

1. *Initialization and groups formation*: In IA, the population of randomly generated possible solutions are divided into some equal-sized political parties. Here, each member of a party is considered a solution. The position of an individual in a political party depends upon his fitness. An individual with the highest fitness in a political party is considered a local party leader. However, the best local party leader is considered a global leader. Individuals in parties compete with others and desire to improve themselves or maintain their current position.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: The individuals in each party are ranked based upon their evaluated fitness as local party leader, second-best individual, ordinary individuals, local second-worst individual, and local worst individual. Each local party leader selects the best improvement using the roulette wheel selection in its fitness calculated after introspection, competing with the second-best and following the global leader to maintain its status in the party. However, ordinary individuals introspect and follow the leader.
  - (b) *Inter-group reproduction(s)*: If the distance between the fitness of the local worst individual and the second-worst individual in each party is higher than a pre-specified value, then the local worst individuals change their ideology by switching to another party. Every other ordinary party individual tries to introspect himself once and follows all the local party leaders with a desire to become one.
3. *Group refreshment*: The ranking of individuals is performed to find the new local best, worst and ordinary individuals.
4. *Termination condition(s)*: IA terminates if there is no significant change in the local party leaders for a significant number of iterations or the maximum number of iterations is reached. The best among all local party leaders provides the best solution.

**Table 20** IA and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in IA
1A. Grouping mechanism	i(a)
1B. Evaluation	(b)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(b) & (c)
2B. Inter-group reproduction(s)	(c), (d) & (e)
3. Group refreshment	(a)
4. Termination criteria	(b) & (c)

#### Algorithm 11 Ideology Algorithm

- 1: Initialize the parameters and generate a random population of individuals
- 2: Create equal-sized political parties
- 3: Evaluate each individual
- 4: **repeat**
- 5:   Rank individuals in each party
- 6:   Search in the neighborhood of each party best, each second party best, and the global best members
- 7:   Worst individuals switch their parties due to a predefined condition
- 8:   Ordinary individuals introspect and follow other local party individuals
- 9:   Update party individuals
- 10: **until** no significant change over iterations or the maximum number of iterations

#### 4.11 Political optimizer (PO)

Political optimizer (PO) (Askari et al. 2020) is a socio-politically inspired meta-heuristic. It mimics all phases of politics, including inter-party elections, constituency/seat allocation, election campaigns, party switching, and parliamentary affairs. The algorithm works on the following premises.

In politics, each party member tries to win the election. Further, each party tries to maximize the number of seats in parliament to form a government. During the electoral process, party members campaign for votes and use their

previous experience to win over the voters. The parties collaborate and compete to win over the voters. Furthermore, candidates follow other candidates and sometimes switch parties.

The algorithmic steps for PO are outlined as follows, and the correspondence of the steps of PO with those of the general framework is provided in Table 21. The pseudocode for PO is provided in Algorithm 12.

1. *Initialization and groups formation*: In PO, a particular case is visualized wherein the number of parties, constituencies, and party members is the same. During initialization, a population of individuals is generated and divided sequentially into several equal-sized political parties. Each of these party members also represents election candidates. The corresponding candidates in various parties contest the elections from the corresponding constituencies. The fitness of each party member is evaluated in a general election, and the fittest member in each party is elected as the party leader. Moreover, the winners from all the constituencies become the constituency winners or parliamentarians.
2. *Reproduction mechanisms*:
  - (a) *Intra-group reproduction(s)*: In the election campaign, a recent past-based position updating strategy (RPPUS), a position updating strategy modeling the learning behaviors of politicians from the previous election, is used. This election campaign updates party members and candidates according to their corresponding party leaders and constituency winners, depending on their new and previous fitness.
  - (b) *Inter-group reproduction(s)*: Each party member is selected with some gradually decreasing probability and swapped/exchanged with the least fit member of a randomly selected party. Furthermore, constituency winners are updated during parliamentary affairs based on another randomly selected constituency winner.
3. *Group refreshment*: After each election, the constituency winners and party leaders are reallocated in the government formation process.
4. *Termination condition(s)*: PO terminates after a maximum number of iterations, and the best party leader is reported as the best solution.

Manita and Korbaa (2020) proposed a binary version of PO using eight transfer functions (Mirjalili and Lewis 2013) categorized into S-shaped and V-shaped to solve feature selection problems on gene expression data. Askari and Younas (2021a) proposed an improved political optimizer (IPO) by enhancing its position-updating mechanism and demonstrated its applicability on benchmark complex

**Table 21** PO and corresponding steps of the general framework

Steps in general framework	Option(s) chosen in PO
1A. Grouping mechanism	i(a)
1B. Evaluation	(b)
1C. Leader selection	(b)
2A. Intra-group reproduction(s)	(a)
2B. Inter-group reproduction(s)	(b) & (e)
3. Group refreshment	(a)
4. Termination criteria	(b)

**Algorithm 12** Political Optimizer

- 1: Initialize the parameters and generate a random population of individuals
- 2: Create equal-sized parties
- 3: Election: evaluate each party member
- 4: Government formation: compute the sets of all party leaders and constituency winners
- 5: **repeat**
- 6: Election campaign: update party members and candidates according to their corresponding party leaders and constituency winners, respectively
- 7: Party switching: exchange members between parties
- 8: Election and government formation
- 9: Parliamentary affairs: constituency winners are updated based upon another randomly selected constituency winner
- 10: **until** the maximum number of iterations

landscapes and engineering optimization problems. Basetti et al. (2021) proposed a quasi-oppositional-based political optimizer (QOPO) by incorporating quasi-opposition-based learning (QOBL) (Tizhoosh 2005) to improve the exploration and convergence capability of the political optimizer and utilized it to solve economic emission load dispatch problem with valve-point loading. Zhu et al. (2021) proposed seven variants of PO with different interpolation and refraction learning strategies. Xu et al. (2022) proposed an improved political optimizer, namely the quantum Nelder–Mead political optimizer (QNMPO), to solve performance optimization in photovoltaic systems. QNMPO uses the quantum rotation gate method to rotate the population of individual solutions and the Nelder–Mead simplex method to improve the solution quality by searching the neighborhood of the best-found solution. Table 22 lists hybrid versions of PO.

A comprehensive view of the similarities and differences between the SIEAs described is provided in Table 23, which shows how steps of each SIEA are mapped into the generic framework by choosing amongst the possible alternatives for each step. The table is designed to get the overall perspective in a single picture. From the table, it is possible to discern precisely where the different SIEAs are similar and how they differ in a pair-wise comparison. Furthermore, Table 24 provides the utilization count of the framework components in

**Table 22** Hybrid Versions of PO

Hybrid Version	Application	Publications
PO + Random Vector Functional Link (RVFL)	Metal cutting process response prediction	Elsheikh et al. (2021)
PO + Feedforward Neural Network (FNN)	FNN training problems	Askari and Younas (2021b)
PO + Loss Sensitivity Factor (LSF)	Distributed Generators and Electric vehicles allocation in a distribution network system	Dharavat et al. (2021)
PO + Voltage Stability Index (VSI)	Capacitor Allocation problem in Distribution Systems	Mani et al. (2021)
PO + Shannon entropy function	Flying Ad Hoc Networks Clustering	Asaamoning et al. (2021)
PO + Adaptive $\beta$ -hill climbing (A $\beta$ HC)	Grey-scale image contrast enhancing	Khan et al. (2022)
PO + Fireworks Algorithm	Numerical and Engineering Optimization	Dong et al. (2022)
PO + Dragonfly Algorithm	Soil Moisture and Heat Level Prediction in IoT	Muppidi et al. (2022)
PO + Multi-verse Optimizer	Tumor Classification and Survival Prediction of Brain Tumor Patients with MRI	Rajeswari et al. (2022)
PO + Sine Cosine Algorithm	Tweet Hashtag Recommendation	Banbhrani et al. (2021)
PO + Sailfish optimization algorithm (SOA)	Optimal DeepMRSeg-based tumor segmentation	Neelima et al. (2022)
PO + Tunicate Swarm Algorithm	UAV (Unmanned Aerial Vehicle) network secure communication	Sangeetha Francelin et al. (2022)
PO + Competitive Multi-verse Optimization enabled Deep Neuro fuzzy network	Student performance estimation in spark environment	Jyoti Baruah and Baruah (2022)

considered SIEAs. The following points emerge based on the data provided in it.

1. The highlighted (in bold) mechanisms serve as the most popular choices in SIEAs. However, the best choice of mechanisms for a new SIEA depends on the combination of mechanisms to balance the exploration and exploitation requirements perfectly.
2. Some SIEA components are yet to be attempted and can be investigated in the future.
3. It might be possible to develop some new and more promising approaches by judiciously selecting other social phenomena and modeling the search on their basis.

## 5 Application areas

In this section, an exhaustive search is performed. The results are given in Table 25, which lists the various reported applications of the state-of-the-art SIEAs. The following salient points can be summarized from a study of the analysis of Table 25.

1. Almost all SIEAs have been applied to solve various real-function optimization problems.
2. Imperialist competitive algorithm and election campaign optimization have been more popular among researchers attempting real-function optimization problems.

3. Society and civilization algorithm, imperialist competitive algorithm, parliamentary optimization, election campaign optimization, group leaders optimization, and soccer league competition algorithm have been applied to solve discrete combinatorial optimization problems.
4. Imperialist competitive algorithm appears to be the most popular SIEA with maximum reported applications considering both combinatorial and real-function optimization problems as surveyed in Hosseini and Khaled (2014).
5. In most cases, the SIEA proposed and published reports better results than traditional EAs.

## 6 Conclusions and future directions

Evolutionary computation has undergone vast and diverse developments in the past few decades. Although the initial inspiration came from biological evolution, some subsequent developments were inspired by the collective intelligence of animals and insects. It is well known that human social evolution has been much faster than biological evolution. Therefore, inspiration from social phenomena, e.g., human behavior exchange and knowledge transfer, has been used to design a new evolutionary computing paradigm dubbed socio-inspired evolutionary algorithms (SIEAs). Numerous SIEAs have been proposed that employ a variety of terminologies picked from the domain of inspiration, viz. elections, and societal and imperial colonization, to name a few. This diverse terminology has created a situation where

**Table 23** Comparison of the state-of-art SIEAs on the basis of algorithmic components

Algo	Grouping mechanism	Evaluation	Leader selection	Intra-group reproduction(s)	Inter-group reproduction(s)	Group updating	Termination criteria
SCA	vi	(a)	(b)	(a)	(a)	(d)	(b)
SLC	i(a)	(a)	(b)	(a), (c) & (d)	(d)	(d)	(b)
SELO	iv(a)	(b)	(c)	(a), (b) & (c)	(a), (b) & (d)	(a)	(b)
NPO	iv(a)	(b)	(c)	(c)	(a)	(a)	(b)
ICA	iii(b)	(a)	(a)	(a) & (c)	(e)	(a) & (b)	(a) & (b)
POA	i(a)	(b)	(b)	(a)	(d)	(a) & (c)	(a)
ECO	iv(a)	(b)	(a)	(a)	(d)	(a)	(b)
GLOA	i(a)	(b)	(b)	(a) & (c)	(d)	(a)	(b)
EA	v	(b)	(a)	(a)	(d)	(a) & (c)	(b)
IA	i(a)	(b)	(b)	(b) & (c)	(c), (d) & (e)	(a)	(b) & (c)
PO	i(a)	(b)	(b)	(a)	(b) & (e)	(a)	(b)

**Table 24** General framework’s steps utilization counts (most popular choices in bold)

Step	Mechanism Utilization count
1A. Grouping mechanism	<b>i(a) 5</b> , i(b) 0, ii(a) 0, ii(b) 0, iii(a) 0, iii(b) 1, iv(a) 3, iv(b) 0, v 1, vi 1
1B. Evaluation	(a) 3, <b>(b) 8</b>
1C. Leader selection	(a) 3, <b>(b) 6</b> , (c) 2
2A. Intra-group reproduction	<b>(a) 9</b> , (b) 2, (c) 6, (d) 1
2B. Inter-group reproduction	(a) 3, (b) 2, (c) 1, <b>(d) 7</b> , (e) 3
3. Group refreshment	<b>(a) 9</b> , (b) 1, (c) 2, (d) 2
4. Termination criterion	(a) 2, <b>(b) 10</b> , (c) 1

**Table 25** Applications of SIEAs

SIEA	Applications	Publications
SCA	Single objective constrained engineering design optimization problems	Akhtar et al. (2002); Ray and Liew (2003)
	Combined heat and power economic dispatch problems	Sharma et al. (2018)
SLC	Systems of nonlinear equations	Moosavian and Roodsari (2013)
	Optimal design of water distribution networks in urban areas	Moosavian and Kasaee Roodsari (2014)
	Knapsack problem	Moosavian (2015)
	Set covering problem	Jaramillo et al. (2016a, b, c)
	Truss Structure Designing	Moosavian and Moosavian (2017)
	Capacitated vehicle routing problem	Anderson (2018)
	Power consumption optimization in wireless sensor networks	Ebrahimi and Tabatabaei (2020)
	Attribute reduction based on rough Set theory	Abdolrazzagh-Nezhad and Adibiyan (2021)
	Load frequency control in nonlinear interconnected power system	Dogan (2021a)
	Concrete strength prediction	Ehsani et al. (2021)
SELO	Pavement maintenance planning of large-scale transportation networks considering energy consumption	Golroo et al. (2021)
	Load flow analysis of power systems	Srilakshmi et al. (2022)
NPO	50 benchmark functions	Kumar et al. (2018)
	Maximizing profits in crop planning	Brindha and Gopi (2019)
NPO	Optimal sizing of a hybrid energy system (HES)	Mohammed et al. (2020a, b, c)

Table 25 continued

SIEA	Applications	Publications
	Zoning map for drought prediction using integrated machine learning models	Mohamadi et al. (2020)
ICA	Test list generation for interaction testing in IoT	Alsewari (2021)
	Nash equilibrium point achievement	Rajabioun et al. (2008)
	Off-line scheduling problem with rejection	Jolai et al. (2010)
	Hybrid flow-shop scheduling	Behnamian and Zandieh (2011)
	Mixed-model U-line balancing and sequencing problem	Lian et al. (2012)
	PID controller designing	Atashpaz-Gargari and Lucas (2007a); Atashpaz Gargari et al. (2008); Atashpaz-Gargari et al. (2008)
	MIMO PIID controller designing	Atashpaz-Gargari et al. (2008)
	Materials property characterization from sharp indentation test	Biabangard-Oskouyi et al. (2009)
	Integrated product mix-outsourcing problem	Nazari-Shirkouhi et al. (2010)
	Optimum skeletal structures designing	Kaveh and Talatahari (2010)
	Linear induction motor designing	Lucas et al. (2010)
	Optimizing the free convection heat transfer in a vertical cavity with flow diverters	Karami et al. (2011)
	Optimal simultaneous coordinated tuning of damping controller	Bijami and Marnani (2012)
	Plate fin heat exchanger designing	Yousefi et al. (2012)
	Adhesive-bonded fiber glass strip optimization	Mozafari et al. (2012)
	Robust PID controller for load-frequency control of power systems	Shabani et al. (2013)
	Sliding mode controller designing	Jalali et al. (2013)
Fractional order PID controller design for LFC in electric power systems	Taher et al. (2014)	
POA	Non-convex economic dispatch problem	Bijami et al. (2014)
	Ground vibration production in quarry blasting	Hajihassani et al. (2015)
	Harmonic minimization in multilevel inverters	Etesami et al. (2015)
	Benchmark numerical functions	Borji (2007); Borji and Hamidi (2009)
	Task scheduling	de Marcos et al. (2010)
	Overlapping community detection in social networks	Altunbey and Alatas (2015)
	Automatic mining of numerical classification rules	Kiziloluk and Alatas (2015)
ECO	Web pages classification	Kiziloluk and Ozer (2017)
	Community structure identification in social networks	Shakya et al. (2020)
	Engineering design problems	He and Zhang (2010)
	Multimodal functions optimization	Lv et al. (2010)
	Optimal tuning PID controller for First Order Lag plus Time Delay System	Wenge et al. (2010a)
	Tuning digital PID controllers for discrete-time systems	Wenge et al. (2010b)
	Multi-peak optimization problems	Wenge et al. (2010c)
	Pressure vessel design problems	Zhang et al. (2011a)
	Generation expansion planning in deregulated power systems	Shayeghi and Esmaeli (2013)
	Re-use reverse logistics system model of disused electric appliances	Xie et al. (2013a)
Quantity prediction and optimal disposal capacity of waste electric appliances	Xie et al. (2013b)	

Table 25 continued

SIEA	Applications	Publications
	Rotated image matching	Zeng et al. (2015)
	Unit commitment problems with wind power effect	Alizadeh et al. (2016)
	Feature points matching	Mei et al. (2016)
	Multilevel Thresholding segmentation technique	TU and LV (2016)
	Fault diagnosis	Xie et al. (2016a)
	Image comparison	Xie et al. (2016b)
	Image matching	Xie et al. (2016c)
	Marbling stone slab image segmentation	Xie et al. (2016d)
	Public transport scheme optimization	Xie et al. (2016e)
	Video tracking	Xie et al. (2016f)
	Stereo matching	Xie et al. (2016g)
	Detecting maximum inscribed rectangle	Xie et al. (2017)
GLOA	Decomposing unitary matrix into a proper-minimum cost quantum gate sequence	Daskin and Kais (2011); Cao et al. (2012); Daskin (2014)
	Job scheduling in a grid computing system	Pooranian et al. (2013)
	QoS and energy consumption aware service composition and optimal-selection in cloud manufacturing systems	Xiang et al. (2014)
	Disruption management for predictable new job arrivals in cloud manufacturing	Liu et al. (2017)
	Economic load dispatch problems	Roy et al. (2017); Shah et al. (2020)
	Low-cost circuit approximations for the Hamiltonian simulation	Dutta et al. (2018)
EA	Benchmark Functions	Emami and Derakhshan (2015)
	Groundwater level prediction	Emami et al. (2018); Emami and Emami (2019); Choopan et al. (2020); Emami et al. (2020)
	Suspended sediment load prediction of rivers	Emami et al. (2019); Emami and Parsa (2022a)
	Random $k$ satisfiability in Hopfield neural network	Abubakar et al. (2020); Sathasivam et al. (2020); Bazuhair et al. (2021); Abubakar et al. (2021)
	Crop cultivation pattern optimization	Choopan and Emami (2020)
	Optimal Economic Water Allocation	Emami and Choopan (2020)
	Concrete Dams cracks evaluation	Emami et al. (2021)
	Concrete Gravity Dam Section Optimization	Emami and Parsa (2022b)
ID	Unconstrained Benchmark Functions	Huan et al. (2017)
PO	PI Controller design	Devarapalli et al. (2020)
	Proton exchange membrane fuel cell (PEMFC) parameters estimation problem	Diab et al. (2020)
	Solar photovoltaic cell unknown parameters identification	Premkumar et al. (2020)
	Levelized Cost of Hybrid Wind-Solar-Diesel-Battery System Optimization	Singh et al. (2020)
	Photovoltaic parameters estimation	Yousri et al. (2020)
	Structural optimization	Awad (2021)
	Optimal allocation of Multiple Distributed Generators and Shunt Capacitors in a Distribution System	Dharavat and SUDABATTULA, S.K., Velamuri, S. (2021)

Table 25 continued

SIEA	Applications	Publications
	Concentric circular antenna arrays (CCAAs) radiation properties improvement	Durmus and Kurban (2021)
	Structural design optimization for buckling-restrained braces (BRBs)	Hoseini et al. (2021)
	Color aerial image multilevel thresholding	Kurban et al. (2021)
	Energy-management system for a microgrid installation	Suresh et al. (2021)
	Integration of DGs in the distribution grids	Tolba and Tulsy (2021)
	Vehicle design optimization	Yıldız et al. (2021)
	Maximum power harvesting technique for offshore permanent magnetic synchronous generator	Zhu et al. (2021)
	Optimal coordination of Directional Overcurrent Relays (DOCRs)	Abdelhamid et al. (2020)
	Optimal adaptive fuzzy management strategy for fuel cell-based DC microgrid	Fathy et al. (2022)
	Solid oxide fuel cell (SOFC) optimal parameters estimation	Fathy and Rezk (2022)
	Energy management strategy for a renewable-based microgrid	Ferahtia et al. (2022)

some algorithms appear to be entirely different but are very similar in matters of significant detail. Anyone trying to understand the area must devote considerable effort to sifting through this terminological maze to arrive at a meaningful understanding of the algorithms and their differences.

In this paper, we present a generalized framework and survey for SIEAs. The proposed framework includes social grouping ideas of a given population into various groups and their two-way evolution. This framework provides a detailed description of each algorithmic component and how it is implemented in practice. A survey of multiple SIEAs is also provided to highlight the working of these algorithms and their improved hybrid versions. Their algorithmic descriptions and pseudocodes are provided. A mapping is provided to the steps outlined in the general framework to help understand the similarities and differences between these methodologies. Various applications of these algorithms found in the literature are also listed for ready reference. Thus, this paper could become an excellent reference as a starting point for anyone interested in this wide upcoming and fascinating field of research.

In future work, it would be interesting to investigate a rigorous study comparing the performance of various SIEAs on a fair platform and diverse applications. Such a study is needed to clarify the relative strengths of the different SIEAs and the respective domains in which each is better than others. Since the different SIEAs appear to have different strengths and search power, further experimental investigations may be made to create a meta-SIEA that encapsulates robust mechanisms in balancing exploration and exploitation of each of

the different SIEAs to perform better than each of them individually on as many problems as possible.

**Author Contributions** All authors contributed equally.

**Funding** The authors declare that no funds, grants, or other support was received during the preparation of this manuscript.

**Data availability** This manuscript has no associated data.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- Abdelhamid M, Kamel S, Mohamed MA, Aljohani M, Rahmann C, Mosaad MI (2020) Political optimization algorithm for optimal coordination of directional overcurrent relays. In: 2020 IEEE electric power and energy conference (EPEC), pp 1–7
- Abdolrazzagh-Nezhad M, Adibiyani A (2021) Attribute reduction based on rough set theory by soccer league competition algorithm. *Nashriyyah-i Muhandisi-i Barq va Muhandisi-i Kampyutar-i Iran* 82(3):169
- Abubakar H, Danrimi ML (2021) Hopfield type of artificial neural network via election algorithm as heuristic search method for random boolean ksatisfiability. *Int J Comput Digital Syst.* <https://doi.org/10.12785/ijcds/100163>
- Abubakar H, Masanawa AS, Yusuf S, Boaku G (2021) Optimal representation to high order random boolean ksatisfiability via election algorithm as heuristic search approach in hopeld neural networks. *J Nigerian Soc Phys Sci.* <https://doi.org/10.46481/jnsps.2021.217>



- Abubakar H, Sabri SRM, Masanawa SA, Yusuf S (2020) Modified election algorithm in hopfield neural network for optimal random k satisfiability representation. *Int J Simul Multidiscip Des Optim* 11:16
- Abubakar H, Sathasivam S (2020) Comparing election algorithm and election campaign optimization algorithm. In: *Aip conference proceedings*, 2266, p 040006
- Abubakar H, Sathasivam S, Alzaeemi SA (2020) Effect of negative campaign strategy of election algorithm in solving optimization problem. *J Qual Measur Anal JQMA* 16(2):171–181
- Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. *Eng Optim* 34(4):341–354
- Alizadeh A, Akbarimajd A, Ghadimi N (2016) Election campaign optimization in unit commitment problem with wind power effect
- Alsewari AA (2021) Nomadic people optimizer for iot combinatorial testing problem. 2021 international conference on intelligent technology, system and service for internet of everything (itss-ioe), pp 1–5
- Altunbey F, Alatas B (2015) Overlapping community detection in social networks using parliamentary optimization algorithm. *Int J Comput Netw Appl* 2(1):12–19
- Anderson JC (2018) Penerapan soccer league competition algorithm untuk menyelesaikan capacitated vehicle routing problem. *Program Studi Teknik Industri Fakultas Teknologi Industri-UNPAR*
- Asaamoning G, Mendes P, Magaia N (2021) A dynamic clustering mechanism with load-balancing for flying ad hoc networks. *IEEE Access* 9:158574–158586
- Askari Q, Younas I (2021) Improved political optimizer for complex landscapes and engineering optimization problems. *Expert Syst Appl* 182:115178
- Askari Q, Younas I (2021) Political optimizer based feedforward neural network for classification and function approximation. *Neural Process Lett* 53(1):429–458
- Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. *Knowl-Based Syst* 195:105709
- Atashpaz-Gargari E, Hashemzadeh F, Lucas C (2008) Designing mimo pid controller using colonial competitive algorithm: applied to distillation column process. *Evolutionary computation, 2008. cec 2008.(iee world congress on computational intelligence). iee world congress on*, pp 1929–1934
- Atashpaz Gargari E, Hashemzadeh F, Rajabioun R, Lucas C (2008) Colonial competitive algorithm: a novel approach for pid controller design in mimo distillation column process. *Int J Intell Comput Cybern* 1(3):337–355
- Atashpaz-Gargari, E, Lucas C (2007a) Designing an optimal pid controller using imperialist competitive algorithm. In: *First joint congress on fuzzy and intelligent systems ferdowsi university of mashhad*, pp 29–31
- Atashpaz-Gargari E, Lucas C (2007b) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *2007 iee world congress on evolutionary computation*, pp 4661–4667. <https://doi.org/10.1109/cec.2007.4425083>
- Awad R (2021) Sizing optimization of truss structures using the political optimizer (po) algorithm. *Structures* (33), 4871–4894
- Bäck T, Fogel DB, Michalewicz Z (1997) *Handbook of evolutionary computation*. CRC Press, Boca Raton
- Banbhrani SK, Xu B, Liu H, Lin H (2021) Sc-political resnet: hashtag recommendation from tweets using hybrid optimization-based deep residual network. *Information* 12(10):389
- Basetti V, Rangarajan SS, Shiva CK, Pulluri H, Kumar R, Collins RE, Senjyu T (2021) Economic emission load dispatch problem with valve-point loading using a novel quasi-oppositional-based political optimizer. *Electronics* 10(21):2596
- Bazuhair MM, Jamaludin SZM, Zamri NE, Kasihmuddin MSM, Mansor M, Alway A et al (2021) Novel hopfield neural network model with election algorithm for random 3 satisfiability. *Processes* 9(8):1292
- Becerra RL, Coello CAC (2006) Cultured differential evolution for constrained optimization. *Comput Methods Appl Mech Eng* 195(33–36):4303–4322
- Behnamian J, Zandieh M (2011) A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Syst Appl* 38(12):14490–14498
- Bi L, Zuo Y, Tao F, Liao TW, Liu Z (2017) Energy-aware material selection for product with multicomponent under cloud environment. *J Comput Inf Sci Eng* 17:3
- Biabangard-Oskouyi A, Atashpaz-Gargari E, Soltani N, Lucas C (2009) Application of imperialist competitive algorithm for materials property characterization from sharp indentation test. *Int J Eng Simul* 10(1):11–12
- Bijami E, Jadidoleslam M, Ebrahimi A, Askari J, Farsangi MM (2014) Implementation of imperialist competitive algorithm to solve non-convex economic dispatch problem. *J Chin Inst Eng* 37(2):232–242
- Bijami E, Marnani JA (2012) Imperialist competitive algorithm for optimal simultaneous coordinated tuning of damping controller. *Int J Tech Phys Prob Eng (IJTPE)* 34–41
- Biswas A, Mishra K, Tiwari S, Misra A (2013) Physics-inspired optimization algorithms: a survey. *J Optim*. <https://doi.org/10.1155/2013/438152>
- Bjorklund DF, Causey K, Periss V (2010) The evolution and development of human social cognition. *Mind the gap* ( 351–371). Springer
- Bonyadi MR, Azghadi S, Hosseini HS (2007) Solving traveling salesman problem using combinatorial evolutionary algorithm. In: *Ifip international conference on artificial intelligence applications and innovations* ( 37–44)
- Borji A (2007) A new global optimization algorithm inspired by parliamentary political competitions. In: *A. Gelbukh & Á.F. Kuri Morales (Eds.), Mica 2007: advances in artificial intelligence* ( 61–71). Berlin, Heidelberg Springer Berlin Heidelberg
- Borji A, Hamidi M (2009) A new approach to global optimization motivated by parliamentary political competitions. *Int J Innovat Comput Inf Control* 5(6):1643–1653
- Brindha GJ, Gopi E (2019) Maximizing profits in crop planning using socio evolution and learning optimization. In: *Socio-cultural inspired metaheuristics* ( 151–174). Springer
- Cao Y, Daskin A, Frankel S, Kais S (2012) Quantum circuit design for solving linear systems of equations. *Mol Phys* 110(15–16):1675–1680
- Choopan Y, Emami S (2020) An approach to reduce water consumption by optimizing and determining of crop cultivation pattern using meta-heuristic algorithms: A case study on moghan plain. *J Appl Res Water Wastewater* 7(1):48–56
- Choopan Y, Emami S, Kheiri Ghojje Bigloo M (2020) Evaluating election, imperialist competitive algorithms and artificial neural network method in investigating the groundwater level of reshtkhar plain. *Amirkabir J Civil Eng* 52(6):1246–1333
- Chung C-J, Reynolds RG (1998) Caep: an evolution-based tool for real-valued function optimization using cultural algorithms. *Int J Artif Intell Tools* 7(03):239–291
- Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surveys (CSUR)* 45(3):1–33
- Cuevas E, Diaz P, Camarena O (2021) Experimental analysis between exploration and exploitation. *Metaheuristic computation: a performance perspective* (249–269). Springer
- Daneshyari M, Yen GG (2004) Talent based social algorithm for optimization. In: *Proceedings of the 2004 congress on evolutionary computation (iee cat. no. 04th8753)* (1, 786–791)

- Darwin C (1859) The origin of species by means of natural selection. Pub One Info
- Darwin C (2018) On the origin of species: Or; the preservation of the favoured races in the struggle for life. Read Books Ltd
- Darwish A (2018) Bio-inspired computing: algorithms review, deep analysis, and the scope of applications. *Future Comput Inf J* 3(2):231–246
- Daskin A (2014) Quantum circuit design methods and applications (Unpublished doctoral dissertation). Purdue University
- Daskin A, Kais S (2011) Decomposition of unitary matrices for finding quantum circuits: application to molecular hamiltonians. *J Chem Phys* 134(14):144112
- Daskin A, Kais S (2011) Group leaders optimization algorithm. *Mol Phys*. <https://doi.org/10.1080/00268976.2011.552444>
- de Marcos L, García A, García E, Martínez JJ, Gutiérrez JA, Barchino R, Otón S (2010) An adaptation of the parliamentary metaheuristic for permutation constraint satisfaction. In: *Ieee congress on evolutionary computation (1–8)*
- Devarapalli R, Lakshmi NJN, Prasad U (2020) Application of a novel political optimization in optimal parameter design of pi controller for the bldc motor speed control. In: *2020 international conference on emerging frontiers in electrical and electronic technologies (ice-foot) (1–6)*
- Dharavat N, Sudabattula SK, Velamuri S (2021) Integration of distributed generation and electric vehicles in a distribution network using political optimizer. In: *2021 4th international conference on recent developments in control, automation & power engineering (rdcape) (521–525)*
- Dharavat N, Sudabattula SK, Velamuri S (2021) Optimal allocation of multiple distributed generators and shunt capacitors in a distribution system using political optimization algorithm. *Int J Renew Energy Res (IJRER)* 11(4):1478–1488
- Diab AAZ, Tolba MA, El-Magd AGA, Zaky MM, El-Rifaie AM (2020) Fuel cell parameters estimation via marine predators and political optimizers. *IEEE Access* 8:16698–167018
- Dogan A (2021) Analysis of soccer league competition algorithm for load frequency control in nonlinear interconnected power system. *Mühendislik Bilimleri ve Tasarım Dergisi* 9(3):866–879
- Dogan A (2021) Optimum siting and sizing of wts, pvs, esss and evcss using hybrid soccer league competition-pattern search algorithm. *Eng Sci Technol Int J* 24(3):795–805
- Dong J, Zou H, Li W, Wang M (2022) Gpofwa: a hybrid greedy political optimizer with fireworks algorithm for numerical and engineering optimization problems. *Research Square*
- Dorigo M, Di Caro G (1999) Ant colony optimization: a new metaheuristic. In: *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406) (2, 1470–1477)*
- Durmus, A, Kurban R (2021) Optimal synthesis of concentric circular antenna arrays using political optimizer. *IETE J Res* 1–10
- Dutta, S, Suau, A, Dutta, S, Roy, S, Behera, BK, Panigrahi, PK (2018) Quantum circuit design methodology for multiple linear regression. *arXiv preprint arXiv:1811.01726*
- Ebrahimi S, Tabatabaei S (2020) Using clustering via soccer league competition algorithm for optimizing power consumption in wsn (wireless sensor networks). *Wireless Pers Commun* 113(4):2387–2402
- Ehsani M, Naseri H, Saeedi Nezhad R, Etebari Ghasbeh M, Moghadas Nejad F (2021) Compressive strength prediction of ordinary concrete, fly ash concrete, and slag concrete by novel techniques and presenting their optimal mixtures. *Amirkabir J Civil Eng* 53(10):1–11
- Eiben AE, Schippers CA (1998) On evolutionary exploration and exploitation. *Fund Inform* 35(1–4):35–50
- Elsheikh AH, Abd Elaziz M, Das SR, Muthuramalingam T, Lu S (2021) A new optimized predictive model based on political optimizer for eco-friendly mql-turning of aisi 4340 alloy with nano-lubricants. *J Manuf Process* 67:562–578
- Emami H (2019) Chaotic election algorithm. *Comput Inf* 38(6):1444–1478
- Emami H, Derakhshan F (2015) Election algorithm: a new socio-politically inspired strategy. *AI Communications* 28 3 591–603
- Emami H, Emami S (2019) Presentating a new approach for evaluating the hydro-geochemical quality of groundwater using swarm intelligence algorithms. *Iranian J Ecohydrol* 6(1):177–190
- Emami H, Emami S, Heydari S (2019) Prediction suspended sediment load of river using meta-heuristic algorithms. *Iranian J Irrig Drainage* 13(5):1426–1438
- Emami H, Lotfi S (2013) Graph colouring problem based on discrete imperialist competitive algorithm. *arXiv preprint arXiv:1308.3784*
- Emami S, Choopan Y (2020) Optimal economic of water allocation using ea and ica evolutionary algorithms. *Amirkabir J Civil Eng* 52(3):601–612
- Emami S, Choopan Y, Parsa J (2018) Modeling the groundwater level of the miandoab plain using artificial neural network method and election and genetic algorithms. *Iranian J Ecohydrol* 5(4):1175–1189
- Emami S, Emami H, Choopan Y, Parsa J, Jahandideh O (2020) Modeling groundwater quality using three novel hybrid support vector regression models. *Adv Environ Technol* 6(2):99–110
- Emami S, Emami H, Parsa J (2022) Performance of vertically cosine shape weir using artificial intelligence. *Model Earth Syst Environ* 8(1):787–798
- Emami S, Parsa J (2022) Optimal design of river groynes using metaheuristic models. *Environ Water Eng* 8(1):146–160
- Emami S, Parsa J (2022) Optimization of concrete gravity dam section using new election meta-heuristic algorithm. *J Struct Constr Eng* 8(12):164–183
- Emami S, Parsa J, Emami H (2021) Evaluating cracks in concrete dams using meta-heuristic algorithms and artificial neural networks. *J Struct Constr Eng* 8(2):194–215
- Etesami M, Farokhnia N, Fathi SH (2015) Colonial competitive algorithm development toward harmonic minimization in multilevel inverters. *IEEE Trans Industr Inf* 11(2):459–466
- Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. *J Water Resour Plan Manag* 129(3):210–225
- Fan X, Sayers W, Zhang S, Han Z, Ren L, Chizari H (2020) Review and classification of bio-inspired algorithms and their applications. *J Bionic Eng* 17(3):611–631
- Fathy A, Ferahtia S, Rezk H, Yousri D, Abdelkareem MA, Olabi A (2022) Optimal adaptive fuzzy management strategy for fuel cell-based dc microgrid. *Energy* 247:123447
- Fathy A, Rezk H (2022) Political optimizer based approach for estimating soft optimal parameters for static and dynamic models. *Energy* 238:122031
- Ferahtia S, Rezk H, Abdelkareem MA, Olabi A (2022) Optimal techno-economic energy management strategy for building's microgrids based bald eagle search optimization algorithm. *Appl Energy* 306:118069
- Fogel DB (1997) The advantages of evolutionary computation. *Bio-computing and emergent computation: Proceedings of bcec97 (1–11)*. World Scientific Press
- Giddings FH (1914) *Inductive sociology: A syllabus of methods, analyses and classifications, and provisionally formulated laws*. Macmillan
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning (1st Ed)*. Boston, MA, USA Addison-Wesley Longman Publishing Co., Inc
- Golroo A, Fani A, Naseri H (2021) Pavement maintenance planning of large-scale transportation networks considering energy consumption. *Amirkabir J Civil Eng* 53(6):25–25

- Gopinath S, Balakrishnan P (2022) A pca-npogbdt strategy-based protection scheme to differentiate between inverter and distribution line faults plus detection and identification of faulty section in microgrid. *Int J Intell Syst* 37(2):1273–1298
- Hajihassani M, Armaghani DJ, Marto A, Mohamad ET (2015) Ground vibration prediction in quarry blasting through an artificial neural network optimized by imperialist competitive algorithm. *Bull Eng Geol Env* 74(3):873–886
- Hayes EC (1911) The classification of social phenomena. *Am J Sociol* 17(1):90–118
- He C, Zhang X (2010) Application of the election campaign optimization algorithm to engineering optimum design problems. In: 2010 international symposium on computational intelligence and design (2, 34–37)
- Hoseini SM, Parastesh H, Hajirasouliha I, Ferdowsi A (2021) Structural design optimization of all-steel buckling-restrained braces using intelligent optimizers. *Int J Steel Struct* 21(6):2055–2070
- Hosseini S, Khaled AA (2014) A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research. *Appl Soft Comput* 24:1078–1094
- Hosseini-Hemati S, Karimi S, Shaeisi GH (2021) How does the civilized gravitational search algorithm solve the optimal dg placement? *J Energy Manag Technol* 5(3):8–20
- Huan TT, Kulkarni AJ, Kanesan J, Huang CJ, Abraham A (2017) Ideology algorithm: a socio-inspired optimization methodology. *Neural Comput Appl* 28(S1):845–876
- Iacoban R, Reynolds RG, Brewster J (2003) Cultural swarms: modeling the impact of culture on social interaction and problem solving. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*. sis'03 (cat. no. 03ex706), pp 205–211
- Jalali A, Piltan F, Keshtgar M, Jalali M (2013) Colonial competitive optimization sliding mode controller with application to robot manipulator. *Int J Intell Syst Appl* 5(7):50
- Jaramillo, A, Crawford, B, Soto, R, Misra, S, Olguín E, Rubio ÁG, Villablanca SM (2016a) An approach to solve the set covering problem with the soccer league competition algorithm. In: *International conference on computational science and its applications*, pp 373–385
- Jaramillo, A, Crawford, B, Soto, R, Villablanca, SM, Rubio, ÁG, Salas, J, Olguín, E (2016b) Solving the set covering problem with the soccer league competition algorithm. In: *International conference on industrial, engineering and other applications of applied intelligent systems*, pp 884–891
- Jaramillo A, Gómez A, Mansilla S, Salas J, Crawford B, Soto R, Olguín E (2016) Using the soccer league competition algorithm to solve the set covering problem. In: *2016 11th Iberian conference on information systems and technologies (cisti)*, pp 1–4
- Jolai F, Sangari MS, Babaie M (2010) Pareto simulated annealing and colonial competitive algorithm to solve an offline scheduling problem with rejection. *Proceed Inst Mech Eng B J Eng Manuf* 224(7):1119–1131
- Jyoti Baruah A, Baruah S (2022) Pfcvmvo: Political fractional competitive multi-verse optimization enabled deep neuro fuzzy network for student performance estimation in spark environment. *Int J Model Simul Sci Comput* 2250050
- Kamaruzaman AF, Zain AM, Yusuf SM, Udin A (2013) Levy flight algorithm for optimization problems—a literature review. *Appl Mech Mater* 421:496–501
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J Global Optim* 39(3):459–471
- Karami A, Yousefi T, Ghashghaei D, Rezaei E (2011) The application of imperialist competitive algorithm in optimizing the free convection heat transfer in a vertical cavity with flow diverters. *Int J Model Optim* 1(4):289
- Kaveh A, Talatahari S (2010) Optimum design of skeletal structures using imperialist competitive algorithm. *Comput Struct* 88(21):1220–1229
- Kavusi M, Khashei Siuki A, Dastourani M (2020) Optimal design of groundwater monitoring network using the combined election-kriging method. *Water Resour Manage* 34(8):2503–2516
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of icnn'95 - international conference on neural networks*, vol 4, pp 1942–1948. <https://doi.org/10.1109/icnn.1995.488968>
- Khan AH, Ahmed S, Bera SK, Mirjalili S, Oliva D, Sarkar R (2022) Enhancing the contrast of the grey-scale image based on meta-heuristic optimization algorithm. *Soft Comput* 1–23
- Khuat TT, Le MH (2016) A survey on human social phenomena inspired algorithms. *Int J Comput Sci Inf Secur* 14:76–81
- Kiziloluk S, Alatas B (2015) Automatic mining of numerical classification rules with parliamentary optimization algorithm. *Adv Electr Comput Eng* 15(4):17–25
- Kiziloluk S, Ozer AB (2017) Web pages classification with parliamentary optimization algorithm. *Int J Software Eng Knowl Eng* 27(03):499–513
- Kiziloluk S, Özer AB (2019) Hybrid parliamentary optimization and big bang-big crunch algorithm for global optimization. *Turkish J Elect Eng Comput Sci* 27(3):1954–1969
- Kumar M, Kulkarni A (2019) Socio-inspired optimization metaheuristics: a review. pp 241–265. Springer. [https://doi.org/10.1007/978-981-13-6569-0\\_12](https://doi.org/10.1007/978-981-13-6569-0_12)
- Kumar M, Kulkarni A, Satapathy S (2018) Socio evolution & learning optimization algorithm: a socio-inspired optimization methodology. *Futur Gener Comput Syst* 81:252–272
- Kurban R, Durmus A, Karakose E (2021) A comparison of novel meta-heuristic algorithms on color aerial image multilevel thresholding. *Eng Appl Artif Intell* 105:104410
- Lian K, Zhang C, Gao L, Shao X (2012) A modified colonial competitive algorithm for the mixed-model u-line balancing and sequencing problem. *Int J Prod Res* 50(18):5117–5131
- Liu M, Yi S, Wen P, Song H (2017) Disruption management for predictable new job arrivals in cloud manufacturing. *J Intell Syst* 26(4):683–695
- Liu Z-Z, Chu D-H, Song C, Xue X, Lu B-Y (2016) Social learning optimization (slo) algorithm paradigm and its application in qos-aware cloud service composition. *Inf Sci* 326:315–333
- Lucas C, Nasiri-Gheidari Z, Tootoonchian F (2010) Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Convers Manage* 51(7):1407–1411
- Lv W, He C, Li D, Cheng S, Luo S, Zhang X (2010) Election campaign optimization algorithm. *Procedia Comput Sci* 1(1):1377–1386
- Mani V, Varma MD, Krishna KV, Khan Z, Sudabattula SK (2021) Hybrid approach to solve capacitor allocation problem in distribution system using political optimizer algorithm. *Int J Eng Res Appl* 11(5):28–35
- Manita G, Korbaa O (2020) Binary political optimizer for feature selection using gene expression data. *Comput Intell Neurosci*. <https://doi.org/10.1155/2020/8896570>
- Mehdinejad M, Mohammadi-Ivatloo B, Dadashzadeh-Bonab R, Zare K (2016) Solution of optimal reactive power dispatch of power systems using hybrid particle swarm optimization and imperialist competitive algorithms. *Int J Electr Power Energy Syst* 83:104–116
- Mei W-b, Lv W-g, Chen K (2016) Research on feature points matching based on election campaign optimization algorithm. *Mech Electr Eng Technol* 04
- Mirhosseini M, Nezamabadi-pour H (2018) Bica: a binary imperialist competitive algorithm and its application in cbr systems. *Int J Mach Learn Cybern* 9(12):2043–2057

- Mirjalili S, Lewis A (2013) S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm Evol Comput* 9:1–14
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24(11):1097–1100
- Mohamadi S, Sammen SS, Panahi F, Ehteram M, Kisi O, Mosavi A, Al-Ansari N (2020) Zoning map for drought prediction using integrated machine learning models with a nomadic people optimization algorithm. *Nat Hazards* 104(1):537–579
- Mohammed AQ, Al-Anbari KA, Hannun RM (2020a) Introducing newly developed nomadic people optimizer (npo) algorithm to find optimal sizing of a hybrid renewable energy. In: *Iop conference series: materials science and engineering*, vol 928, p 022052
- Mohammed AQ, Al-Anbari KA, Hannun RM (2020b) Multi-objective npo minimizing the total cost and co 2 emissions for a stand-alone hybrid energy system. In: *International conference of reliable information and communication technology*, pp 351–363
- Mohammed AQ, Al-Anbari KA, Hannun RM (2020) Optimal combination and sizing of a stand-alone hybrid energy system using a nomadic people optimizer. *IEEE Access* 8:200518–200540
- Moosavian N (2015) Soccer league competition algorithm for solving knapsack problems. *Swarm Evol Comput* 20:14–22
- Moosavian N, Kasaei Roodsari B (2014) Soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm Evol Comput* 17:14–24
- Moosavian N, Moosavian H (2017) Testing soccer league competition algorithm in comparison with ten popular meta-heuristic algorithms for sizing optimization of truss structures. *Int J Eng* 30(7):926–936
- Moosavian N, Roodsari BK et al (2013) Soccer league competition algorithm, a new method for solving systems of nonlinear equations. *Int J Intell Sci* 4(1):7–16
- Moscato P (1989) On evolution, search, optimization, gas and martial arts: toward memetic algorithms (Tech. Rep.). CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826
- Mousavirad, S, Ebrahimpour-Komleh H (2013) Feature selection using modified imperialist competitive algorithm. *Iccke 2013*, pp 400–405
- Mozafari H, Abdi B, Ayob A (2012) Optimization of adhesive-bonded fiber glass strip using imperialist competitive algorithm. *Procedia Technol* 1:194–198
- Muppidi S, PG OP, et al (2022) Dragonfly political optimizer algorithm-based rider deep long short-term memory for soil moisture and heat level prediction in iot. *Comput J*
- Nazari-Shirkouhi S, Eivazy H, Ghodsi R, Rezaie K, Atashpaz-Gargari E (2010) Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm. *Expert Syst Appl* 37(12):7615–7626
- Neelima G, Chigurukota DR, Maram B, Girirajan B (2022) Optimal deepmrseg based tumor segmentation with gan for brain tumor classification. *Biomed Signal Process Control* 74:103537
- Neme A, Hernández S (2009) Algorithms inspired in social phenomena. In: *Nature-inspired algorithms for optimisation*, pp 369–387. Berlin, Heidelberg Springer Berlin Heidelberg
- Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: a literature review. *Swarm Evol Comput* 2:1–14
- Norman MG, Moscato P et al (1991) A competitive and cooperative approach to complex combinatorial search. In: *Proceedings of the 20th informatics and operations research meeting*, pp 3–15
- Nozarian S, Soltanpoora H, Jahanb MV (2012) A binary model on the basis of imperialist competitive algorithm in order to solve the problem of knapsack 1-0. In: *Proceedings of the international conference on system engineering and modeling*, pp 130–135
- Osmani A, Mohasefi JB, Gharehchopogh FS (2022) Sentiment classification using two effective optimization methods derived from the artificial bee colony optimization and imperialist competitive algorithm. *Comput J* 65(1):18–66
- Pei J, Song Q, Liao B, Liu X, Pardalos PM (2021) Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration. *Ann Oper Res* 298(1):407–444
- Pekel E, Soner Kara S (2017) Passenger flow prediction based on newly adopted algorithms. *Appl Artif Intell* 31(1):64–79
- Pooranian Z, Shojafar M, Abawajy JH, Singhal M (2013) Gloa: a new job scheduling algorithm for grid computing. *Int J Interact Multimed Artif Intell* 2(1):59–64
- Premkumar M, Sowmya R, Jangir P, Kumar JS (2020) A new and reliable objective functions for extracting the unknown parameters of solar photovoltaic cell using political optimizer algorithm. In: *2020 international conference on data analytics for business and industry: way towards a sustainable economy (icdabi)*, pp 1–6
- Qiao Y, Dao T-K, Pan J-S, Chu S-C, Nguyen T-T (2020) Diversity teams in soccer league competition algorithm for wireless sensor network deployment problem. *Symmetry* 12(3):445
- Rajabioun R, Atashpaz-Gargari E, Lucas C (2008) Colonial competitive algorithm as a tool for nash equilibrium point achievement. In: *International conference on computational science and its applications*, pp 680–695
- Rajeswari R, Neelima G, Maram B, Angadi A (2022) Mvpo predictor: Deep learning-based tumor classification and survival prediction of brain tumor patients with mri using multi-verse political optimizer. *Int J Pattern Recognit Artif Intell* 225 2006
- Ramezani F, Lotfi S (2013) Social-based algorithm (sba). *Appl Soft Comput* 13(5):2837–2856
- Rashedi E, Nezamabadi-Pour H (2007) Gravitational search algorithm (Unpublished doctoral dissertation). M.Sc. Thesis, Shahid Bahonar University of Kerman, Kerman, Iran
- Ray T, Liew K (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Trans Evol Comput* 7(4):386–396
- Rechenberg I (1970) Optimierung technischer systeme nach prinzipien der biologischen evolution (Unpublished doctoral dissertation). Technische Universität, Fakultät für Maschinenwissenschaft
- Reynolds RG (1994) An introduction to cultural algorithms. In: *Proceedings of the third annual conference on evolutionary programming*, pp 131–139
- Roy S, Bhattacharjee K, Bhattacharya A (2017) A modern approach to solve of economic load dispatch using group leader optimization technique. *Int J Energy Optim Eng (IJEEOE)* 6(1):66–85
- Salih SQ, Alsewari AA (2020) A new algorithm for normal and large-scale optimization problems: nomadic people optimizer. *Neural Comput Appl* 32(14):10359–10386
- Sangeetha Francelin VF, Daniel J, Velliangiri S (2022) Intelligent agent and optimization-based deep residual network to secure communication in uav network. *Int J Intell Syst*
- Sathasivam S, Mansor M, Kasihmuddin MSM, Abubakar H et al (2020) Election algorithm for random k satisfiability in the hopfield neural network. *Processes* 8(5):568
- Selvakumar AI, Thanushkodi K (2009) Optimization using civilized swarm: solution to economic dispatch with multiple minima. *Electric power systems research* 79(1):8–16
- Senadji B, Dawes L (2010) Effect of group formation on performance, task management and social loafing. In: *Proceedings of the 21st annual conference for the australasian association for engineering education*, pp 64–69
- Shabani H, Vahidi B, Ebrahimpour M (2013) A robust pid controller based on imperialist competitive algorithm for load-frequency control of power systems. *ISA Trans* 52(1):88–95
- Shah K, Bhattacharjee K, Godwal S (2020) Solving of economical load dispatch using efficient group leader optimization technique. In: *Technologies for sustainable development: Proceedings of the 7th*

- nirma university international conference on engineering (nuicone 2019), november 21–22, 2019, Ahmedabad, p 181
- Shakya HK, Shaik N, Singh K, Sinha G, Biswas B (2020) Community structure identification in social networks inspired by parliamentary political competitions. *Data management, analytics and innovation*, pp 445–457. Springer
- Sharma E, Sharma S, Sachdev A (2018) Combined heat and power economic dispatch using society civilization algorithm. *Int J Sci Tech Adv* 4(1):109–112
- Shayeghi H, Esmaeli H (2013) Eco based solution for generation expansion planning in deregulated power systems. *Electr Electron Eng* 3(2):33–42
- Siddique N, Adeli H (2017) Nature-inspired computing: physics-and chemistry-based algorithms. Chapman and Hall/CRC
- Singh P, Pandit M, Srivastava L (2020) Optimization of leveled cost of hybrid wind-solar-diesel-battery system using political optimizer. In: 2020 IEEE first international conference on smart technologies for power, energy and control (stpec), pp 1–6
- Srilakshmi K, Babu PR, Venkatesan Y, Palanivelu A (2022) Soccer league optimization for load flow analysis of power systems. *Int J Numer Model Electron Networks Devices Fields* 35(2):e2965
- Suresh V, Jasinski M, Leonowicz Z, Kaczorowska D, Reddy KH et al (2021) Political-optimizer-based energy-management system for microgrids. *Electronics* 10(24):3119
- Taher SA, Fini MH, Aliabadi SF (2014) Fractional order pid controller design for lfc in electric power systems using imperialist competitive algorithm. *Ain Shams Eng J* 5(1):121–135
- Tao F, Bi L, Zuo Y, Nee A (2016) A hybrid group leader algorithm for green material selection with energy consideration in product design. *CIRP Ann* 65(1):9–12
- Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (cimca-iawtic'06), vol 1, pp 695–701
- Tolba MA, Tulsy VN (2021) Integration of dgs optimally to enhance the voltage profile and stability index of distribution grid. In: 2021 3rd international youth conference on radio electronics, electrical and power engineering (reepe), pp 1–6
- Trivers R, Dawkins R (1976) *The selfish gene*. Oxford University Press, Oxford, UK
- TU X-a, & LV W-g (2016) Multilevel thresholding segmentation technique based on election campaign optimization algorithm. *Mech Electr Eng Technol* 03
- Varki A, Altheide TK (2005) Comparing the human and chimpanzee genomes: searching for needles in a haystack. *Genome Res* 15(12):1746–1758
- Wang Y, Zhang Y, Tao F, Chen T, Cheng Y, Yang S (2019) Logistics-aware manufacturing service collaboration optimisation towards industrial internet platform. *Int J Prod Res* 57(12):4007–4026
- Wenge L, Deyuan L, Siyuan C, Shaoming L, Zeyu C (2010a) Optimal tuning pid controller for first order lag plus time delay system by election campaign optimization algorithm. In: 2010 international conference on electrical and control engineering, pp 1430–1433
- Wenge L, Deyuan L, Siyuan C, Shaoming L, Zeyu C (2010b) Tuning digital pid controllers for discrete-time system by election campaign optimization algorithm. 2010 international conference on mechanic automation and control engineering, pp 2559–2562
- Wenge L, Zhiyong L, Deyuan L, Siyuan C, Shaoming L, Xiangwei Z, Zeyu C (2010c) Election campaign optimization algorithm for multi-peak optimization problems. In: 2010 third international conference on information and computing, vol 3, pp 154–157
- Whitley D, Gordon VS, Mathias K (1994) Lamarckian evolution, the baldwin effect and function optimization. In: International conference on parallel problem solving from nature, pp 5–15
- Xiang F, Hu Y, Yu Y, Wu H (2014) Qos and energy consumption aware service composition and optimal-selection based on pareto group leader algorithm in cloud manufacturing system. *CEJOR* 22(4):663–685
- Xie Q, Lv W, Liu Z, Zhang X, Luo S, Cheng S (2010) Constrained optimization with election campaign algorithm. In: 2010 the 2nd international conference on industrial mechatronics and automation, vol 1, pp 370–373
- Xie Q, Zhang X, Lv W, Cheng S (2016a) Extension neural network optimized by election campaign algorithm for fault diagnosis. In: 2016 6th international conference on mechatronics, computer and education informationization (mcei 2016)
- Xie Q, Zhang X, Lv W, Cheng S (2016b) Image compared by election campaign algorithm. In: 2016 6th international conference on management, education, information and control (meici 2016)
- Xie Q, Zhang X, Lv W, Cheng S (2016c) Image matching based on election campaign algorithm. In: 2016 6th international conference on management, education, information and control (meici 2016), pp 1244–1248
- Xie Q, Zhang X, Lv W, Cheng S (2016d) Marbling stone slab image segmentation based on election campaign algorithm. In: 2016 6th international conference on mechatronics, computer and education informationization (mcei 2016)
- Xie Q, Zhang X, Lv W, Cheng S (2016e) Public transport scheme optimized by election campaign algorithm and extension theory. In: 2016 6th international conference on mechatronics, computer and education informationization (mcei 2016)
- Xie Q, Zhang X, Lv W, Cheng S (2016f) Video tracking based on election campaign algorithm. In: 2016 6th international conference on mechatronics, computer and education informationization (mcei 2016)
- Xie QH, Zhang XW, Lv WG, Cheng SY (2016g) Stereo matching based on election campaign algorithm. In: *Matec web of conferences*, vol 63, p 01031
- Xie Q-H, Zhang X-W, Lv W-G, Cheng S-Y (2017) Detecting maximum inscribed rectangle based on election campaign algorithm. In: 2016 6th international conference on advanced design and manufacturing engineering (icadme 2016), pp 815–820
- Xie QH, Zhang XW, Lv WG, Cheng SY, Huang HX, Cai SD (2013a) Research on re-use reverse logistics network of disused electric appliances based on election campaign algorithm. *Appl Mech Mater* 397:1121–1128
- Xie QH, Zhang XW, Lv WG, Cheng SY, Lin CT, Yang YJ (2013b) Quantity prediction and optimal disposal capacity of waste electric appliances based on election campaign algorithm. *Appl Mech Mater* 411:2395–2399
- Xu B, Heidari AA, Kuang F, Zhang S, Chen H, Cai Z (2022) Performance optimization of photovoltaic systems: Reassessment of political optimization with a quantum nelder-mead functionality. *Sol Energy* 234:39–63
- Yıldız BS, Pholdee N, Bureerat S, Erdaş MU, Yıldız AR, Sait SM (2021) Comparison of the political optimization algorithm, the archimedes optimization algorithm and the levy flight algorithm for design optimization in industry. *Mater Test* 63(4):356–359
- Yousefi M, Darus A, Mohammadi H (2012) An imperialist competitive algorithm for optimal design of plate-fin heat exchangers. *Int J Heat Mass Transf* 55(11):3178–3185
- Yousri D, Abd Elaziz M, Oliva D, Abualigah L, Al-qaness MA, Ewees AA (2020) Reliable applied objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study. *Energy Convers Manage* 223:113279
- Zargarnataj M (2007) New election algorithm based on assistant in distributed systems. *Computer systems and applications*, 2007. In: *aiccsa'07. IEEE/ACS international conference on*, pp 324–331
- Zeng Z-m, Lv W-g, Xie Q-h (2015) Rotated image matching based on election campaign algorithm. *Mech Electr Eng Technol* 09

- Zhang H, Lv WG, Cheng SY, Luo SM, Zhang XW (2011a) Election campaign optimization algorithm for design of pressure vessel. *Adv Mater Res* 308:15–20
- Zhang L, Lv W, Wang C, Cheng S, Luo S (2011b) Parameter design and performance study on election campaign optimization algorithm. In: *Computer science and network technology (iccsnt), 2011 international conference on*, vol 1, pp 263–266
- Zhu A, Gu Z, Hu C, Niu J, Xu C, Li Z (2021) Political optimizer with interpolation strategy for global optimization. *PLoS ONE* 16(5):e0251204

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)